FULL-ATOM PEPTIDE DESIGN WITH DUAL-FLOW CONTRASTIVE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Peptides can bind protein surfaces with high specificity and engage interfaces that are typically hard to drug, making them a highly promising class of therapeutics. Recent advances in structure-based peptide design leverage diffusion and flow-matching models to simultaneously generate binding peptide sequences and structures. While existing methods can optimize binding affinities through property guidance or preference alignment, they require extensive model retraining or auxiliary classifiers, and still struggle with poor side-chain modeling and severe steric clashes. To address these limitations, we introduce a lightweight and transferable approach that concurrently optimizes binding energy and side-chain conformations. Specifically, we propose PepContra, a dual-flow contrastive guidance method for full-atom peptide co-design. This contrastive guidance integrates dual flows during sampling, leveraging their difference to drive the generative process toward high-quality regions and away from suboptimal spaces. We further optimize the joint sampler by synchronizing discrete amino-acid types with continuous side-chain torsions at each Euler step. Experimental results demonstrate that PepContra outperforms existing approaches across co-design benchmarks, achieving superior binding affinities and highly competitive stereochemistry.

1 INTRODUCTION

Peptides are short amino acid chains connected by amide bonds, with typically fewer than 50 residues. Compared with small molecules, peptides offer larger contact areas and flexible backbones that adapt to flat surfaces such as protein–protein interfaces, and therefore bind with higher specificity (Fosgerau & Hoffmann, 2015). At the same time, their smaller size and lower immunogenicity allow them to penetrate tissues and to be obtained by chemical synthesis. Consequently, the development of therapeutic peptide drugs remains a critical direction in drug discovery.

Generative models have demonstrated significant progress in drug design. Compared with conventional methods that need costly screening or manually structure modification, deep generative models built on diffusion (Song et al., 2021) or flow matching (Lipman et al., 2023) frameworks can generate candidate binders in an end to end manner (Yim et al., 2023; Campbell et al., 2024; Guan et al., 2023a). However, designing peptide binders remains difficult because their conformational flexibility creates a tight coupling between sequence and structure—any inconsistency among these components can severely compromise binding quality. To address this dilemma, recent advances propose to jointly model full-atom peptide sequence and structure (Li et al., 2024; Lin et al., 2024; Kong et al., 2025b; Qian et al., 2026), which referred to as sequence-structure co-design.

Despite these promising advancements, several difficulties in peptide co-design still remain. First, the conformation and binding of designed peptide still fall short of native peptides. Steric clashes and misfolded side chains are usually observed. Second, to optimize model performance, most existing methods apply property-driven guidance or alignment targeted at one or a few optimization objectives (Gu et al., 2024; Cheng et al., 2025; Zhou et al., 2024; Nguyen et al., 2026). Every additional objective, however, demands finetuning or retraining the model with extra modules, which is far from convenient. It is also argued that different classifiers often share little common sampling distribution, resulting optimizing one objective degrades another (Zhou et al., 2024).

To address these challenges, we introduce **PepContra**, a contrastive-guided multimodal flow matching method for full-atom peptide design. Our method builds on the following hypothesis: the dis-

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.

tribution sampled by a sufficiently trained generative model already covers high-quality data region, but because conditional flow sampling gravitates toward the mean of the conditional distribution, sparse high-quality regions receive very low probability (Karras et al., 2024). Rather than swapping in a new distribution with external data, or defining an explicit classifier for every property, PepContra simply make the high-quality points the model already covers more likely to be sampled. Specifically, we first trains a peptide design base model built on multimodal flow matching. To construct the guiding flow, we then finetune two models, HQ and LQ, on self-synthesized data. During sampling, the generated structure and sequence are continually pushed toward the ideal distribution by the HQ flow and away from the region of the LQ flow. Additionally, we propose synchronized sampling, which binds the state of the side-chain torsions to the amino acid type when updating to resolve the inconsistency between sequence and side-chain sampling. Our contributions can be summarized as follows:

- We propose PepContra, a novel contrastive-guided flow matching method for peptide sequence-structure co-design, which fully exploits the potential of an existing base model without large-scale retraining or modification.
- We propose synchronized sampling, which improves the accuracy of side-chain modeling and mitigates the collapse of amino acid types in sequence generation.
- PepContra achieves superior performance on co-design task, particularly strong on interface energy and stereochemistry.

2 Related Works

2.1 Generative models for peptide design

In recent years, deep generative models have made significant progress in peptide design. Early efforts focused primarily on sequence design using protein language models (Nijkamp et al., 2023). For target-specific generation, PepMLM (Chen et al., 2026) leverages the structural knowledge encoded in ESM-2 (Lin et al., 2023) to design peptide sequences with only the target sequence as input. Recognizing that a peptide's function relies heavily on conformational flexibility, the field has largely shifted toward sequence and 3D structure co-design. Recent advances tackle this challenge through diverse generative frameworks. One line of work (Li et al., 2024; Lin et al., 2024; Wu et al., 2025a) directly models the 3D structure and sequence of the peptide using conditional flow matching models. Another line of work, represented by Kong et al. (2024), employs latent diffusion models (Rombach et al., 2022) to learn a joint latent space of peptide sequences and structures. Kong et al. (2025b) further extends this approach to design binders of multiple molecular domains within a single model. Given that co-design models usually suffer from information shortcuts and inconsistency between joint modeling and inference, Qian et al. (2026) and Wu et al. (2025b) utilize Bayesian flow networks (Graves et al., 2025) for each peptide modality and then combine them with a rationalized inference process to achieve better co-design performance.

2.2 Property guidance and preference alignment in binder design

The ultimate goal of binder design is to generate molecules with specific properties, so as to bind a given receptor or perform a desired function. In addition to learning the property distribution implicitly from the data, researchers have also proposed a range of methods that guide and align generative models to achieve this goal. Inspired by Diffusion-DPO (Wallace et al., 2024), one intuitive idea introduces a preference alignment mechanism, which finetunes the model so that preferred molecules are generated directly. Gu et al. (2024), Cheng et al. (2025) and Zhou et al. (2024) take key metrics such as binding energy, as the reward, and construct preference pairs to perform alignment during training. ProteinDPO (Widatalla et al., 2026) instead takes wet-lab stability measurements as the preference to optimize a protein language model.

A second line of thought originates from classifier guidance (Dhariwal & Nichol, 2021), which pushes the generative model to sample towards the preference by incorporating the gradient of an additionally defined classifier. DecompDiff (Guan et al., 2023b) introduces penalty terms on connectivity and on clashes at sampling time. ConfDiff (Wang et al., 2024) instead uses the energy and forces of a molecular force field to correct the score of a score-based model. PepTri (Nguyen

et al., 2026) performs gradient guidance on latent vector from energy, evolutionary information, and mutual information while PepMimic (Kong et al., 2025a) exploits the guidance in latent space by defining the distance to the binding interface to be mimicked.

3 METHODS

In this section we will present dual-flow contrastive guidance for full-atom peptide design. We first define the notation used throughout this paper. In Section 3.2 we offer both theoretical (albeit non-formal) justification and an intuitive understanding of the motivation behind dual flows. In Section 3.3 we introduce the specific implementation of the proposed contrastive guidance on Euclidean space, the $\mathrm{SO}(3)$ manifold, the flat torus of side-chain torsions, and the categorical probability of residue types. Finally, in Section 3.4 we explain how to sample with dual-flow contrastive guidance and synchronization.

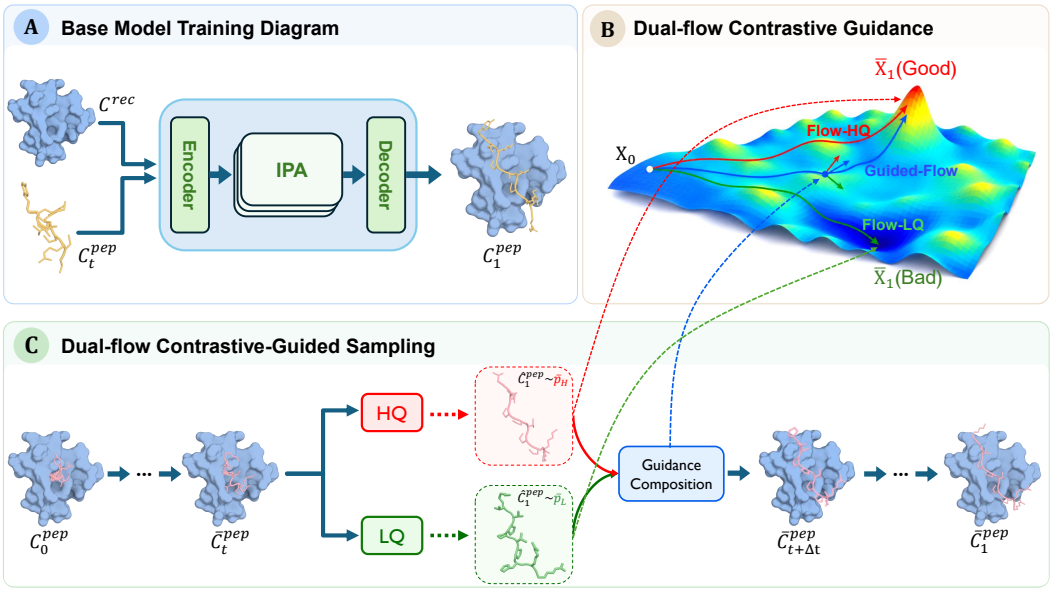


Figure 1: The overview of PepContra. **(A)** Training diagram of base model: The interpolated peptide sequence and structure are fed into the encoder-decoder based flow-matching model with the pocket as condition to predict the native sequence and structure. **(B)** Conceptual illustration of dual-flow contrastive guidance. At every point in the guided-flow path, the vector field is the composition of flow-HQ and flow-LQ. **(C)** Sampling with contrastive flow guidance. HQ and LQ models are finetuned from base model in **(A)**.

3.1 NOTATIONS

We focus on designing both the sequence and the full-atom structure of a peptide $\mathcal{C}^{\mathrm{pep}}$ that binds a given receptor $\mathcal{C}^{\mathrm{rec}}$. Following Jumper et al. (2021) a residue can be characterized by its amino acid type, rigid backbone frame, and side-chain torsion. The frame is constructed from the four backbone heavy atoms N–C_α–C–O with C_α at the origin, and is parameterized by a translation $x \in \mathbb{R}^3$ and a rotation $R \in \mathrm{SO}(3)$ with respect to the global reference frame (Yim et al., 2023). The side chain is flexible compared with the rigid backbone and is described by up to four rotatable torsion angles; together with the backbone torsion that governs the position of the O atom, we collect them into $\chi \in [0, 2\pi)^5$. The type $a \in \{1, \ldots, S\}$ with $S = 20$ represents the canonical amino acid types. An n-residue peptide is thus written as $\mathcal{C}^{\mathrm{pep}} = \{(x^j, R^j, \chi^j, a^j)\}_{j=1}^n$. The goal of peptide co-design is formalized as modeling the conditional distribution $q(\mathcal{C}^{\mathrm{pep}} \mid \mathcal{C}^{\mathrm{rec}})$.

3.2 Motivation and Derivation of Dual Flows

Peptide binders have intrinsic properties and correlations between structure and sequence that the neural network cannot easily and perfectly learn. Thus we reasonably assume that the flow model corresponding to the real data distribution $q(\mathcal{C}^{\text{pep}})$ is always unavailable. We further assume that the synthesized data in which peptides are generated by the base model follow a distribution of the form $\tilde{p}(\mathcal{C}^{\text{pep}}) = q(\mathcal{C}^{\text{pep}}) \exp(-\epsilon(\mathcal{C}^{\text{pep}}))$, where $\epsilon(\cdot)$ is a function for which we do not know the explicit formula. This assumption implies certain constraints on $\epsilon(\cdot)$, since $\tilde{p}$ is self-normalized, thereby satisfying the property required of a probability density.

We curate two distinct datasets using the self-sampled peptides generated by the base flow model. Specifically, we use the pretrained base model to generate peptides for the receptors in the *training* split and score them with two offline quality filters. See Appendix A.2 for the details of the filters. We slightly finetune the pretrained base model on the two datasets separately to derive two flow models, denoted FLOW-LQ (low quality) and FLOW-HQ (high quality), and we write their induced distributions as

$$p_L(\mathcal{C}^{\text{pep}}) = q(\mathcal{C}^{\text{pep}}) \exp\big(-\epsilon_L(\mathcal{C}^{\text{pep}})\big), \qquad p_H(\mathcal{C}^{\text{pep}}) = q(\mathcal{C}^{\text{pep}}) \exp\big(-\epsilon_H(\mathcal{C}^{\text{pep}})\big). \tag{1}$$

Because the two self-synthesized datasets explicitly differ in data quality, we could assume that for any desirable peptide $\mathcal{C}_H$ we have $p_H(\mathcal{C}_H) > p_L(\mathcal{C}_H)$, and for any undesirable peptide $\mathcal{C}_L$ we have $p_H(\mathcal{C}_L) < p_L(\mathcal{C}_L)$. Although we have no knowledge about the explicit formulas of $\epsilon_L(\cdot)$ and $\epsilon_H(\cdot)$, the above claims still indicate that $\epsilon_L(\mathcal{C}_H) > \epsilon_H(\mathcal{C}_H)$ and $\epsilon_L(\mathcal{C}_L) < \epsilon_H(\mathcal{C}_L)$. The *difference* between the two flows is therefore informative about quality even though neither flow is accurate.

This inspires us to sample high-quality peptides by utilizing the difference between these two flows, as shown in Figure 1. Our goal is to sample following the probability defined by

$$\bar{p}(\mathcal{C}^{\text{pep}}) \;\propto\; p_H(\mathcal{C}^{\text{pep}}) \left[\frac{p_H(\mathcal{C}^{\text{pep}})}{p_L(\mathcal{C}^{\text{pep}})}\right]^{\gamma} \;=\; q(\mathcal{C}^{\text{pep}}) \exp\big(\gamma\, \epsilon_L(\mathcal{C}^{\text{pep}}) - (1+\gamma)\, \epsilon_H(\mathcal{C}^{\text{pep}})\big), \tag{2}$$

where $\gamma > 0$ is a constant hyperparameter. It is straightforward to prove that $\bar{p}(\mathcal{C}_H) > q(\mathcal{C}_H)$ and $\bar{p}(\mathcal{C}_L) < q(\mathcal{C}_L)$, which means that the calibrated target distribution $\bar{p}$ puts more (resp. less) probability mass on desirable (resp. undesirable) samples than the real-data underlying distribution q. Therefore, sampling according to equation 2 leads to high-quality peptides even when all the models we utilize perform worse than the ideal generator corresponding to the ground-truth data distribution.

3.3 Dual Flow Contrastive Guidance

We now show how to sample according to the distribution in equation 2. We start with the flow over Euclidean space and then carry the construction to the remaining three geometries.

Euclidean Space. The relationship between probability-flow ODEs, whose drift is a vector field, and score-based generative models, whose drift is a score function, follows from the Fokker–Planck equation (Song et al., 2021), and has also been made explicit for flow matching (Zheng et al., 2023). In our case we can likewise relate the vector field over Euclidean space to the underlying score function,

$$v_t^{\text{pos}}\big(\mathcal{C}_t^{\text{pep}}\big) = \frac{1}{t}\, x_t + \frac{1-t}{t}\, \nabla_{x_t} \log p_t\big(\mathcal{C}_t^{\text{pep}}\big). \tag{3}$$

We define the calibrated marginal probability path at time t as

$$\bar{p}_t\big(\mathcal{C}_t^{\text{pep}}\big) \;\propto\; p_{H,t}\big(\mathcal{C}_t^{\text{pep}}\big) \left[p_{H,t}\big(\mathcal{C}_t^{\text{pep}}\big) / p_{L,t}\big(\mathcal{C}_t^{\text{pep}}\big)\right]^{\gamma}, \tag{4}$$

where $p_{H,t}$ (resp. $p_{L,t}$) is the marginal probability path of FLOW-HQ (resp. FLOW-LQ) at t. Applying the score function to equation 4 gives

$$\nabla_{x_t} \log \bar{p}_t = (1+\gamma)\, \nabla_{x_t} \log p_{H,t} - \gamma\, \nabla_{x_t} \log p_{L,t}, \tag{5}$$

and we accordingly define the calibrated vector field as

$$\bar{v}^{\text{pos}}\big(\mathcal{C}_t^{\text{pep}}\big) \;:=\; (1+\gamma)\, v_H^{\text{pos}}\big(\mathcal{C}_t^{\text{pep}}\big) - \gamma\, v_L^{\text{pos}}\big(\mathcal{C}_t^{\text{pep}}\big), \tag{6}$$

where v_H^{pos} (resp. v_L^{pos}) is the vector field of FLOW-HQ (resp. FLOW-LQ), and we ignore the residue index j for brevity. By plugging the linear relationship equation 3 between the vector field and the score function into both sides, we can derive that $\bar{v}^{\text{pos}}$ generates the marginal probability $\bar{p}_t$, which justifies using the calibrated vector field to sample approximately from the calibrated target distribution.

Notably, equation 6 resembles the formula of classifier-free diffusion guidance (Ho & Salimans, 2022), with the unconditional branch replaced by a lower-quality flow. Likewise, it also holds a straightforward meaning: standing on a data point $\mathcal{C}_t^{\text{pep}}$, the vector field of FLOW-HQ is calibrated by adjusting its direction away from the vector field of FLOW-LQ, because the latter points more to the regions of undesirable samples. The mechanism behind this motivates us to name the method contrastive guidance.

In practice we calibrate the estimated clean samples, i.e., the predicted $\hat{x}_1$, instead of the vector fields directly. Plugging equation 15 into equation 6, we can easily derive that

$$\bar{v}^{\text{pos}}\big(\mathcal{C}_t^{\text{pep}}\big) = \frac{(1+\gamma)\hat{x}_1^H - \gamma\hat{x}_1^L - x_t}{1-t}, \tag{7}$$

which implies that calibrating the vector fields of FLOW-HQ and FLOW-LQ as in equation 6 is equivalent to directly calibrating the predicted clean sample as

$$\bar{x}_1\big(\mathcal{C}_t^{\text{pep}}\big) := (1+\gamma)\hat{x}_1^H - \gamma\hat{x}_1^L = \hat{x}_1^L + (1+\gamma)\big(\hat{x}_1^H - \hat{x}_1^L\big), \tag{8}$$

where $\hat{x}_1^H$ (resp. $\hat{x}_1^L$) is the frame translation of the clean sample predicted by FLOW-HQ (resp. FLOW-LQ) given the same noisy peptide $\mathcal{C}_t^{\text{pep}}$. Guidance therefore amounts to extrapolating along the segment that runs from the low-quality prediction through the high-quality one, which transfers to every remaining modality.

$\mathrm{SO}(3)$ **Manifold.** We follow the above intuition to calibrate the vector field for the rotation of residue frames, extending equation 8 from Euclidean space to the $\mathrm{SO}(3)$ manifold by replacing the straight segment with the geodesic through the two predicted rotations,

$$\bar{R}_1\big(\mathcal{C}_t^{\text{pep}}\big) := \exp_{\hat{R}_1^L}\Big((1+\gamma)\log_{\hat{R}_1^L}\big(\hat{R}_1^H\big)\Big), \qquad \bar{v}^{\text{ori}}\big(\mathcal{C}_t^{\text{pep}}\big) := \frac{\log_{R_t}\big(\bar{R}_1(\mathcal{C}_t^{\text{pep}})\big)}{1-t}, \tag{9}$$

where $\hat{R}_1^H$ (resp. $\hat{R}_1^L$) is the frame rotation of the clean sample predicted by FLOW-HQ (resp. FLOW-LQ). The geometric understanding behind equation 9 is that FLOW-LQ subtly pushes FLOW-HQ away along the geodesic path connecting their predicted $\hat{R}_1$, and the calibrated target remains on the manifold by construction, which a linear combination of rotation matrices would not.

Flat Torus of Side-chain Torsions. Side-chain conformation is where undesirable samples are most easily produced and most costly, since a single rotamer placed in the wrong well is enough to drive a clash with the receptor, and it is also the modality on which the base model has the least reliable signal. Because $\mathbb{T}^5$ inherits its metric from $\mathbb{R}^5$ up to periodicity, the extrapolation of equation 8 carries over once displacements are taken in the minimal-length representative given by equation 21:

$$\bar{\chi}_1\big(\mathcal{C}_t^{\text{pep}}\big) := \Big(\hat{\chi}_1^L + (1+\gamma)\,\mathrm{wrap}\big(\hat{\chi}_1^H - \hat{\chi}_1^L\big)\Big) \bmod 2\pi, \quad \bar{v}^{\text{ang}}\big(\mathcal{C}_t^{\text{pep}}\big) := \mathrm{wrap}\left(\frac{\bar{\chi}_1(\mathcal{C}_t^{\text{pep}}) - \chi_t}{1-t}\right), \tag{10}$$

where $\hat{\chi}_1^H$ (resp. $\hat{\chi}_1^L$) is the torsion vector of the clean sample predicted by FLOW-HQ (resp. FLOW-LQ). With the wrapped form, FLOW-LQ instead pushes FLOW-HQ away along the shorter arc joining their predicted torsions, exactly as in equation 9.

Categorical Probability of Residue Types. The contrastive guidance for the sequence is different from the continuous cases in that there is no vector field or score function acting on the discrete variable itself. Because Appendix D lifts the type onto the logit space, however, the flow that carries it is an ordinary Euclidean flow, and equation 8 applies to the predicted clean logits verbatim,

$$\bar{s}_1\big(\mathcal{C}_t^{\text{pep}}\big) := (1+\gamma)\hat{s}_1^H - \gamma\hat{s}_1^L, \qquad \bar{v}^{\text{type}}\big(\mathcal{C}_t^{\text{pep}}\big) := \frac{\bar{s}_1(\mathcal{C}_t^{\text{pep}}) - s_t}{1-t}, \tag{11}$$

where $\hat{s}_1^H$ (resp. $\hat{s}_1^L$) is the clean logit predicted by FLOW-HQ (resp. FLOW-LQ). What makes this more than a formal substitution is that the guidance applied to the logits is exactly the guidance one would want to apply to the categorical distribution it represents. Writing $\hat{p}_H(a_1 \mid \cdot) = \mathrm{softmax}(\hat{s}_1^H)$ and $\hat{p}_L(a_1 \mid \cdot) = \mathrm{softmax}(\hat{s}_1^L)$ for the clean categoricals predicted by the two flows, the softmax of equation 11 satisfies

$$\mathrm{softmax}\big(\bar{s}_1\big)[i] \;=\; \frac{\hat{p}_H(a_1 = i \mid \cdot)^{1+\gamma}\, \hat{p}_L(a_1 = i \mid \cdot)^{-\gamma}}{\sum_{i'=1}^{S} \hat{p}_H(a_1 = i' \mid \cdot)^{1+\gamma}\, \hat{p}_L(a_1 = i' \mid \cdot)^{-\gamma}} \;=\; \bar{p}(a_1 = i \mid \cdot), \tag{12}$$

which follows by noting that $\hat{p}_H[i]^{1+\gamma}\hat{p}_L[i]^{-\gamma} = \exp\big((1+\gamma)\hat{s}_1^H[i] - \gamma \hat{s}_1^L[i]\big) \cdot Z_H^{-(1+\gamma)} Z_L^{\gamma}$, where the partition functions Z_H and Z_L do not depend on i and therefore cancel under normalization. Hence the calibrated categorical $\bar{p}(a_1 \mid \cdot)$ is the geometric extrapolation of the two predicted categoricals, which is the same tilt as equation 2 applied to the predicted clean type and the discrete counterpart of equation 8 rather than an additional assumption.

Model Architecture. The model architecture of PepContra is a standard encoder–decoder design. We use a geometric equivariant network to embed the target structure and sequence as the generation context. The encoder produce residue representations h and pair representations z, which are fed into the decoder together with the noisy peptide state to predict the vector fields of the generated peptide (details in Appendix B). The final loss is composed of four CFM objectives for their corresponding modalities and two auxiliary losses for the backbone positions and torsion angles, which are weighted and summed as:

$$\mathcal{L} = \mathbb{E}_{t,\, \mathcal{C}_1^{\mathrm{pep}},\, \mathcal{C}_t^{\mathrm{pep}}} \left[\frac{1}{n} \sum_{j=1}^{n} \sum_{k \in \mathcal{M}} \lambda_k \, \mathcal{L}_k^j \right], \qquad \mathcal{M} = \{\mathrm{trans}, \mathrm{rot}, \mathrm{tor}, \mathrm{type}, \mathrm{bb}, \mathrm{ang}\}. \tag{13}$$

3.4 GUIDED AND SYNCHRONIZED SAMPLING

Contrastive Guided Sampling. For dual-flow contrastive guided sampling, we employ FLOW-HQ and FLOW-LQ on the same state $\mathcal{C}_t^{\mathrm{pep}}$, combine their predicted clean peptides through equations 8–12, and take the Euler step with the calibrated fields (see Figure 1). We employ three different values of γ across four modalities where the backbone rotation and translation share the same γ_{bb}. The cost for contrastive flow guidance is two decoder forward passes per step against a single receptor encoding. Compared with classifier guidance methods, we do not involve a property predictor, energy function, or reward model at sampling time; the entire steering signal is the contrast between two short finetunes of one checkpoint.

Synchronized Sampling. We remove the inconsistency between side-chain rotamer prediction and amino-acid type update by bonding the torsion state to the residue type. We denote $\mathcal{J}(a) \subseteq \{1, \ldots, 5\}$ for the torsion slots that a type a possesses and $m(a) \in \{0, 1\}^5$ for the corresponding mask. At every step the existence of a slot is decided by the currently predicted clean type $\bar{a}_1$ alone, and the torsion state that is advanced under one type is never shared with another type. When the predicted type at a position changes, the torsions of the abandoned type are not zeroed but frozen, together with the time $t_{\mathrm{last}}^{(b)}$ and the endpoint $\hat{\chi}_1^{(b)}$ at which that type was last present. If the type reappears n_t steps later at time t', we do not resume from the current state; instead we replay the whole missing interval in a single geodesic update with the last state we restore:

$$\chi_{t'}^{(b)} := \Big(\chi_{last}^{(b)} + \tfrac{t' - t_{\mathrm{last}}^{(b)}}{1 - t_{\mathrm{last}}^{(b)}} \, \mathrm{wrap}\big(\hat{\chi}_1^{(b)} - \chi_{last}^{(b)}\big) \Big) \bmod 2\pi, \qquad b = \bar{a}_1, \tag{14}$$

which covers in one shot the n_t steps the frozen state has missed. Every torsion is therefore read at the same time as the sequence that owns it and stays on a single probability path, which is precisely the regime the model was trained in. We show the sampling process in Algorithm 3, with the original Algorithm 2 given for comparison.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. We curated a training set of protein–peptide complexes from the Protein Data Bank (Berman et al., 2000), PepBDB (Wen et al., 2019) and Q-BioLiP (Wei et al., 2024). Receptor sequences are clustered with MMseqs2 at 40% identity and 80% coverage (Steinegger & Söding, 2017). We evaluate our model on the LNR (Tsaban et al., 2022) dataset, containing 93 held-out complexes. To avoid receptor homology with the test set, each LNR receptor is searched against the clustered training receptors and the entire cluster of the best hit is removed. The total number of complexes used for model training is 9701. See Appendix A for more dataset details.

Evaluation Metrics. Following common practice in peptide co-design (Kong et al., 2025b; Li et al., 2024), we evaluate model performance from peptide geometry, binding affinity, rationality, and diversity. See Appendix E for implementation details. **Amino-acid recovery (AAR)** is the SequenceMatcher identity with the native sequence. **C-RMSD** and **L-RMSD** are peptide C_α RMSD after aligning on the pocket and on the peptide, respectively. **DockQ** (Basu & Wallner, 2016) measures peptide–pocket docking and interface quality compared with the native peptide. **Interface ΔG** is Rosetta `dG_separated` after neighborhood FastRelax (Alford et al., 2017). **IMP** is the fraction of the targets on which the generated peptide outperforms the native binder in terms of ΔG. **JSD$_{\mathbf{bb}}$** and **JSD$_{\mathbf{sc}}$** compare backbone and side-chain dihedrals with those of training peptides. **Sequence diversity** is the fraction of Needleman–Wunsch (Needleman & Wunsch, 1970) clusters at identity 0.6; **Structural diversity** uses C_α RMSD clusters at 2 Å. **Clash** counts all heavy-atom pairs closer than $r_i + r_j - 1.5$ Å per peptide residue.

Baselines. Our baselines include **PepFlow** (Li et al., 2024), **PepGLAD** (Kong et al., 2024), **UniMoMo** (Kong et al., 2025b), **ProBayes** (Wu et al., 2025b), **PepContra-base** is the pretrained model using synchronized sampling without any guidance (Training details in Appendix B.2). For **PepContra** we apply both dual-flow guidance and synchronized sampling (see Appendix G for hyperparameters). We provide detailed baseline implementations in Appendix F.

4.2 PEPTIDE SEQUENCE-STRUCTURE CO-DESIGN

Main Results. We evaluate PepContra on sequence-structure co-design task. For every model we sample 100 designs per target on the test set with 200 Euler steps. We assess the model along four axes: **Recovery** and **Energy** in Table 1, **Rationality** and **Diversity** in Table 2.

PepContra is the best or the second-best method on almost every metric, and its advantage is most pronounced on the energy terms, where it scores clearly better than the competing models. This holds even against UniMoMo, which is pretrained on a large amount of additional cross-domain data such as SAbDab (Dunbar et al., 2014) and CrossDocked (Francoeur et al., 2020). The strong energy results match what we expect from constructing HQ-data and LQ-data and sampling contrastively. The best result on IMP suggests that during sampling the model is pushed towards the low-energy region covered by HQ-data and away from the high-energy region covered by LQ-data, which substantially raises the probability of drawing a low-energy peptide. We provide a further analysis of the generated results in Appendix C.

DockQ, in contrast, does not improve appreciably. The results follow accordingly: because fine-tuning on HQ-data and LQ-data barely separates FLOW-HQ and FLOW-LQ from the base model in terms of DockQ, DockQ gains less from contrastive sampling.

PepContra also performs well on the rationality metrics. The Ramachandran plot in Figure 2 shows that our model captures the backbone torsion distribution of the native peptides. One point deserves attention: UniMoMo is not trained on the same data as our model, and its lead on the structural metrics comes from pretraining on a much larger corpus. With fewer data used for training, our model still attains the second-lowest peptide-internal clash. The side-chain dihedral distances (JSD$_{\mathrm{sc}}$) also improve markedly over PepFlow. It is worth noting that contrastive guidance reduces sequence diversity considerably. This is again a consequence of guiding the sampling stage: as the sampling probability concentrates on the regions with better binding energy, the model becomes far less likely to draw peptides that are diverse in sequence but worse in structure and energy. Diversity is moreover

Table 1: Recovery and empirical energy on LNR.

Method	Recovery				Energy	
	AAR ↑	C-RMSD ↓	L-RMSD ↓	DockQ ↑	ΔG ↓	IMP ↑
PepFlow	0.487	2.07	1.19	0.507	−27.24	23.7%
PepGLAD	**0.506**	3.06	1.57	0.408	−18.67	11.8%
UniMoMo	0.505	2.40	1.11	0.429	−31.10	29.0%
ProBayes	0.489	2.18	**0.84**	0.502	−26.48	22.6%
PepContra-base	0.497	**1.83**	1.12	**0.522**	−28.89	25.8%
PepContra	0.432	2.36	1.22	0.514	**−32.00**	**35.5%**

Table 2: Rationality and diversity on LNR. Clash reports the number of clashing heavy-atom pairs per peptide residue (in: peptide-internal; out: peptide–pocket).

Method	Rationality				Diversity	
	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓	JSD_{bb} ↓	JSD_{sc} ↓	Seq. ↑	Struct. ↑
PepFlow	0.705	0.533	0.223	0.614	0.283	0.408
PepGLAD	2.745	0.715	0.524	0.396	**0.648**	**0.768**
UniMoMo	**0.025**	**0.134**	0.259	**0.186**	0.622	0.623
ProBayes	1.869	0.625	**0.142**	0.805	0.113	0.219
PepContra-base	0.435	0.635	0.168	0.295	0.534	0.280
PepContra	0.152	0.529	0.217	0.337	0.209	0.176

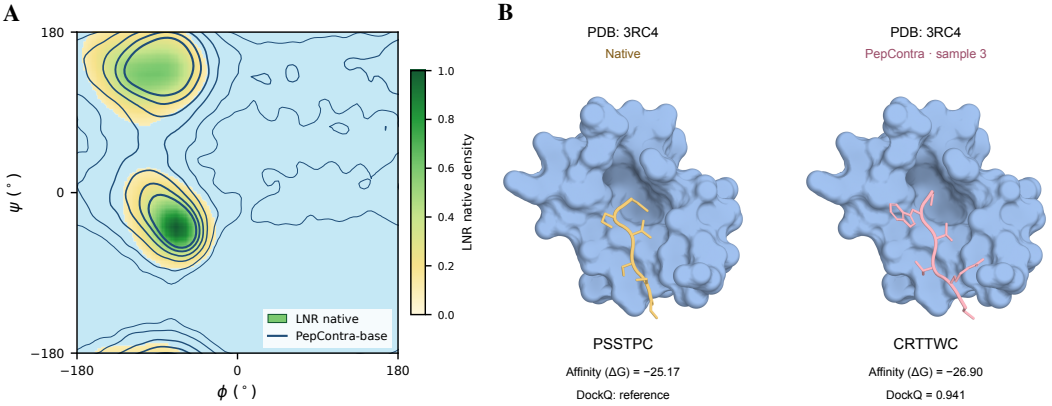


Figure 2: (A) Ramachandran plot of the peptide backbone torsions on LNR. Filled contours show the density of the native peptides; the lines show the distribution generated by PepContra-base. (B) A binder designed for PDB 3RC4. The native peptide and a PepContra sample are shown in the same pocket, together with their sequences, interface ΔG, and DockQ.

tied directly to the capability of the base model. We only change the probability with which the sampler draws high-quality rather than low-quality samples, without modifying the distribution itself, so the performance of the base model sets the ceiling for PepContra. We provide more visualization in Appendix H.

4.3 Ablation Study

We conduct the following ablations: PepContra without synchronized sampling and contrastive flow guidance (**PepContra w/o sync+contra**); PepContra without contrastive flow guidance (**PepContra w/o contra**); $\gamma = 0$, which uses the HQ model only (**HQ-only**); $\gamma = -1$, which uses the LQ model only (**LQ-only**); and the inverse model (**Inverse-PepContra**), which swaps the HQ and LQ models in contrastive guidance. The results in Table 3 show that HQ and LQ differ mainly in energy, in the fraction of good samples, and in the quality of the peptide's own conformation, which agrees with the strategy we used to construct the two sets. Synchronized sampling corrects the peptide-internal

conformation substantially, but contributes little to energy. Dual-flow contrastive guidance steers the model towards structures with more reasonable stereochemistry, and increases both the quality and the frequency of the samples that bind better.

Table 3: Ablation study on the peptide co-design task. We sample 40 peptides for each LNR target.

Method	DockQ ↑	ΔG ↓	IMP ↑	Clash_{in} ↓
PepContra	0.514	$\mathbf{-29.51}$	**28.0%**	**0.149**
PepContra w/o sync+contra	0.507	−25.43	19.4%	0.721
PepContra w/o contra	0.519	−25.51	20.4%	0.447
HQ-only	0.516	−28.61	26.9%	0.283
LQ-only	**0.527**	−22.39	17.2%	0.922
Inverse-PepContra	0.526	−18.83	17.2%	2.542

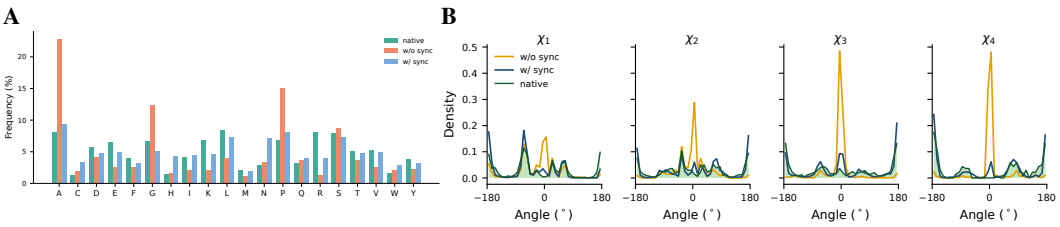


Figure 3: (A) Amino-acid composition of the native peptides and of the peptides generated without and with synchronized sampling. (B) Distributions of χ_1–χ_4 for the same three settings.

Sequence and Side-chain Performance. We further investigate how synchronized sampling improves side-chain conformation and sequence prediction by examining how closely the generated sequences and side-chain torsions follow the distributions of the native peptides. As shown in Figure 3, the original sampling procedure (Li et al., 2024) makes the predicted sequence and the predicted side-chain torsions collapse. In the composition of panel A, the model produces an abnormally high proportion of Ala and Gly, whose side chains are short and carry no side-chain rotamer. The torsion distributions in panel B corroborate this: for all four torsion angles, the original sampler produces a high proportion of rotamers near $0°$, a value that only occur as placeholder for rotamers that do not exist in that amino-acid type during training. Our synchronized sampling alleviates this phenomenon to a large extent, and the fraction of each amino-acid type in the generated sequence is closer to the native statistics.

5 Conclusion

The work introduce a dual-flow contrastive guidance method for structure-based full-atom peptide design. PepContra derives a pair of deliberately unequal flows from a single pretrained base model by finetuning on its own samples, and turns the difference between the two flows into a steering signal on every modality of a peptide residue during sampling. Through guided flow together with a synchronization when updating the structure and sequence within each Euler step, PepContra achieves convincing performance in binding energy and stereochemistry without any explicit energy function or reward model at sampling time, and exhibits potential applications as a general sampling framework for other flow-based binder design model. Nevertheless, PepContra faces limitations in diversity during generation, stemming from the concentration of sampling probability on the low-energy regions covered by the high-quality flow. Furthermore, since guidance only reweights what the base generator can already reach, the capability of the pretrained model still sets the ceiling of our method. The quality filter that defines the two finetuning sets remains handcrafted, both of which represent areas for improvement. Despite these considerations, PepContra stands out as a lightweight and general tool for improving generative models in computational peptide design.

ETHICS STATEMENT

This work concerns computational peptide design. Generated sequences could in principle be applied to harmful bioactive molecules; we intend the method for therapeutic and methodological use, and we will not release protocols aimed at increasing pathogenicity.

USE OF LARGE LANGUAGE MODELS

In this work, we use generative AI tools to

- polish grammar and wording across drafts.
- assist with coding and debugging. Some implementations in our work are assisted with LLMs. We also use several independent debug agents to check the generated code and results.
- collect output results and organize documents. We use some agents to help us monitor the experiment process in case any errors occur. The output results are also collected and arranged into documents by agents.

We have not used generative AI tools to

- generate figures and results. We claim that figures and visualization result are based on our real data and code. The main figure is manually created and the peptide elements come from our training and sampling data.
- generate ideas. The idea and motivation are neither given by agents nor discussed with agents.
- generate references and citations. We have manually verified every reference against its published source.

REFERENCES

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, 2017.

Sankar Basu and Björn Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11(8):e0161879, 2016.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

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 *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=kQwSbv0BR4.

Leo Tianlai Chen, Zachary Quinn, Madeleine Dumas, Christina Peng, Lauren Hong, Moises Lopez-Gonzalez, Alexander Mestre, Rio Watson, Sophia Vincoff, Lin Zhao, et al. Target sequence-conditioned design of peptide binders using masked language modeling. *Nature Biotechnology*, 44(6):1002–1010, 2026.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet,

J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 54787–54819. Curran Associates, Inc., 2024. doi: 10.52202/079017-1738. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/62a58f2130894e44e8a272c563a2c6f1-Paper-Conference.pdf.

Xiwei Cheng, Xiangxin Zhou, Yuwei Yang, Yu Bao, and Quanquan Gu. Decomposed direct preference optimization for structure-based drug design. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=dwSpo5DRk8.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 139054–139084. Curran Associates, Inc., 2024. doi: 10.52202/079017-4413. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fadec8f2e65f181d777507d1df69b92f-Paper-Conference.pdf.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

Keld Fosgerau and Torsten Hoffmann. Peptide therapeutics: current status and future directions. *Drug discovery today*, 20(1):122–128, 2015.

Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. 3d convolutional neural networks and a crossdocked dataset for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200, 2020.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f0d629a734b56a642701bba7bc8bb3ed-Paper-Conference.pdf.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks, 2025. URL https://arxiv.org/abs/2308.07037.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. Aligning target-aware molecule diffusion models with exact energy optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 44040–44063. Curran Associates, Inc., 2024. doi: 10.52202/079017-1398. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4ddfe69f164eae70abc86f0f9cbed7e8-Paper-Conference.pdf.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=kJqXEPXMsE0.

Jiaqi Guan, Xiangxin Zhou, Yuwei Yang, Yu Bao, Jian Peng, Jianzhu Ma, Qiang Liu, Liang Wang, and Quanquan Gu. DecompDiff: Diffusion models with decomposed priors for structure-based drug design. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11827–11846. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/guan23a.html.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11575–11596. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.647. URL https://aclanthology.org/2023.acl-long.647/.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, 2021.

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.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems 37*, pp. 52996–53021, 2024. doi: 10.52202/079017-1679. URL https://doi.org/10.52202/079017-1679.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-atom peptide design with geometric latent diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 74808–74839. Curran Associates, Inc., 2024. doi: 10.52202/079017-2379. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/88ad9774ffcb7a272457e9396f793a07-Paper-Conference.pdf.

Xiangzhe Kong, Rui Jiao, Haowei Lin, Ruihan Guo, Wenbing Huang, Wei-Ying Ma, Zihua Wang, Yang Liu, and Jianzhu Ma. Peptide design through binding interface mimicry with pepmimic. *Nature biomedical engineering*, pp. 1–16, 2025a.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31397–31418. PMLR, 13–19 Jul 2025b.

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 of *Proceedings of Machine Learning Research*, pp. 27615–27640. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24o.html.

Haitao Lin, Odin Zhang, Huifeng Zhao, Dejun Jiang, Lirong Wu, Zicheng Liu, Yufei Huang, and Stan Z. Li. PPFLOW: Target-aware peptide design with torsional flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30510–30528. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24z.html.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

Ngoc-Quang Nguyen, Jaeyoon Jung, Seijung Kim, Sunkyu Kim, and Jaewoo Kang. Peptri: Tri-guided all-atom diffusion for peptide design via physics, evolution, and mutual information. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yQlTgHo1um.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Hao Qian, Shikui Tu, and Lei Xu. Full-atom peptide design via riemannian–euclidean bayesian flow networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24918–24926, 2026.

Pierre H. Richemond, Sander Dieleman, and Arnaud Doucet. Categorical sdes with simplex diffusion, 2022. URL https://arxiv.org/abs/2210.14784.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46495–46513. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/stark24b.html.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Tomer Tsaban, Julia K Varga, Orly Avraham, Ziv Ben-Aharon, Alisa Khramushin, and Ora Schueler-Furman. Harnessing protein folding neural networks for peptide–protein docking. *Nature communications*, 13(1):176, 2022.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Yan Wang, Lihao Wang, Yuning Shen, Yiqun Wang, Huizhuo Yuan, Yue Wu, and Quanquan Gu. Protein conformation generation via force-guided SE(3) diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56835–56859. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wang24cv.html.

Hong Wei, Wenkai Wang, Zhenling Peng, and Jianyi Yang. Q-biolip: A comprehensive resource for quaternary structure-based protein–ligand interactions. *Genomics, Proteomics and Bioinformatics*, 22(1):qzae001, 2024.

Zeyu Wen, Jiahua He, Huanyu Tao, and Sheng-You Huang. Pepbdb: a comprehensive structural database of biological peptide–protein interactions. *Bioinformatics*, 35(1):175–177, 2019.

Talal Widatalla, Ashir A Borah, Samuel H King, Claudia L Driscoll, Rafael Rafailov, and Brian L Hie. Aligning protein-generative models to experimental fitness with proteindpo. *Nature Methods*, pp. 1–9, 2026.

Fang Wu, Zhengyuan Zhou, Shuting Jin, Xiangxiang Zeng, Jure Leskovec, and Jinbo Xu. Surface-based molecular design with multi-modal flow matching. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3192–3203, 2025a.

Hanlin Wu, Yuxuan Song, Zhe Zhang, Zhilong Zhang, Hao Zhou, Wei-Ying Ma, and Jingjing Liu. Rationalized all-atom protein design with unified multi-modal bayesian flow. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 3019–3050. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0105. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0463ec87d0ac1e98a6cbe3d95d4e3e35-Paper-Conference.pdf.

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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40001–40039. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/yim23a.html.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making, 2023. URL https://arxiv.org/abs/2311.13443.

Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 120861–120891. Curran Associates, Inc., 2024. doi: 10.52202/079017-3840. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/daef77101ba5711084a57442c8cf2709-Paper-Conference.pdf.

A Dataset Details

A.1 Data Statistics on Training and Test Set

The complexes used for training come from PDB (Berman et al., 2000), PepBDB (Wen et al., 2019) and Q-BioLiP (Wei et al., 2024). We cluster the receptor sequences with MMseqs2 at 40% identity and 80% coverage (Steinegger & Söding, 2017), which produces 1642 clusters. The largest cluster contains 658 complexes.

The test set is the LNR dataset (Tsaban et al., 2022), containing 93 complexes. To prevent leakage of test receptors into training, every LNR receptor is searched against the 9849 training receptors PDB names and clusters, and the entire cluster of the hit is removed. The largest cluster is unchanged and still contains 658 complexes. The set used for training therefore contains 9701 complexes and covers 8534 PDB entries.

Figure 4 compares peptide length in training and test sets. Training peptides length range from 3 to 25 residues, with the median at length 9. LNR peptides run from 3 to 27, and the single peptide of length 27 is outside the training range.

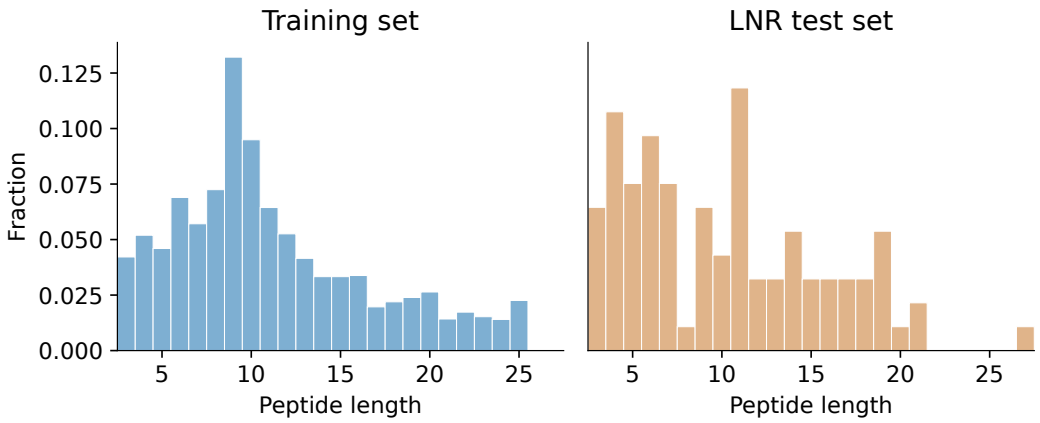


Figure 4: Peptide-length histograms of the 9701 training complexes and the 93 LNR test complexes. Bins are one residue wide, and the vertical axis is the fraction of complexes in each set.

A.2 Data for Training Dual-flow

We synthesize pocket–peptide complexes by self-sampling the training pockets with the base model. We sample 10 peptides for each of the 9701 targets, producing $97{,}010$ candidates in total. High-quality (HQ) and low-quality (LQ) sets are then taken from this pool by successive filters. We observed that short peptides usually obtain a higher DockQ than longer ones, but a worse interface energy (see Figure 5), so the two sets do not share the same filtering for peptide in different length.

For HQ, we keep peptides with DockQ > 0.8, relax the neighborhood, and score ΔG. On the 3376 targets with $\Delta G < -30$ REU we retain at most two lowest-energy peptides per target, which leaves 6209 complexes, and then drop any peptide with an intra-residue clash.

LQ is built in two parts, as we observed that the binding energy and interface of the sampled peptides tend to follow different distributions depending on the peptide length, as shown in Figure 5. The first part keeps a native-like interface with poor energy: DockQ > 0.8 and $\Delta G > -5$ REU (7039 peptides), then we keep at most two worst-energy peptides per target. The second part restores longer peptides, which the first part barely contains. The filter is: length at least 9, DockQ < 0.4, AAR < 0.4, and at least one internal clash. Table 4 records the resulting counts.

Table 4: Statistics of HQ and LQ sets. Length bins are 1–8 / 9–15 / $\geq$ 16. ΔG is the median.

Set	Complexes	Targets	Length bins	Median ΔG
HQ	3944	2656	852 / 2441 / 651	−41.90
LQ, part 1	3863	2409	2798 / 951 / 114	+10.90
LQ, part 2	2477	1439	0 / 868 / 1609	−12.23
LQ, total	6340	3839	2798 / 1819 / 1723	+0.64

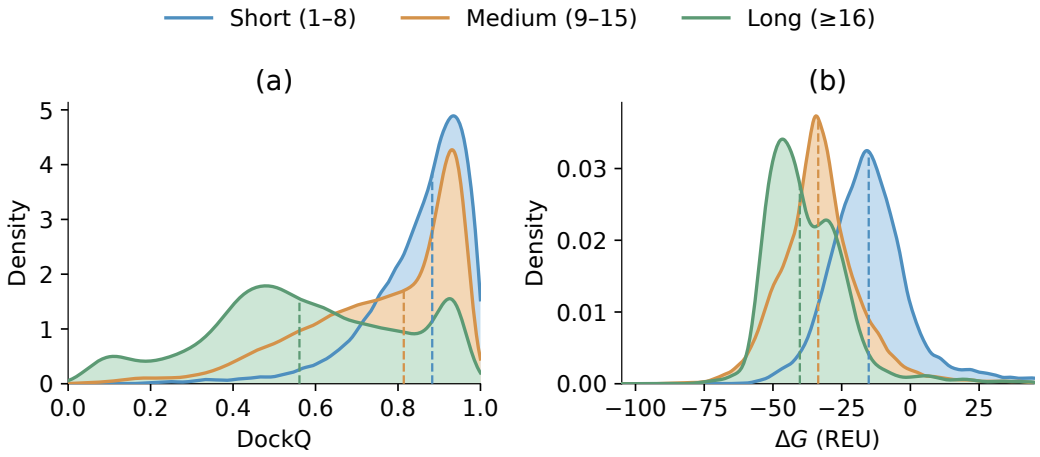


Figure 5: Length-stratified distributions of the self-sampled peptides. (a) DockQ on all 97,010 candidates. (b) Interface energy of the DockQ > 0.8 subset. Dashed lines mark the median of each length group.

B Additional Implementation Details for PepContra

Here, we introduce the implementation details for PepContra.

B.1 Model Details

Encoder. The encoder reads the three-dimensional structure of the target protein and returns a residue (node) embedding together with a residue-pair (edge) embedding. Each residue embedding is built from four inputs, and each input is passed through its own multi-layer perceptron (MLP):

- a learnable embedding of the amino-acid type;
- coordinates of both the backbone atoms and the side-chain atoms;
- sinusoidal embeddings of the backbone dihedral angles;
- sinusoidal embeddings of the side-chain torsion angles.

The four encoded vectors are concatenated and mapped by one further MLP to obtain the final residue embedding. Pair embeddings are formed in the same way from four features:

- a learnable embedding 20×20 entries for the pair of residue types;
- a learnable embedding of the relative sequence offset between the two residues;
- the distance between the two residues;
- the relative orientation, computed from inter-residue backbone dihedral angles and encoded with sinusoidal embeddings.

These pair features are likewise encoded by separate MLPs, concatenated, and passed through another MLP to produce the edge embedding. The encoder is $\mathrm{SE}(3)$-invariant: any global rigid motion

of the input leaves the embeddings unchanged. The residue embedding has width 128, and the pair embedding has width 64.

Decoder. The flow decoder is conditioned on the receptor node and edge embeddings, on the peptide descriptors at the current interpolated state, and on the timestep t. Its backbone is a stack of Invariant Point Attention (IPA) blocks (Jumper et al., 2021). IPA consumes the features above together with the backbone frames, and the attention update is invariant, so it can capture contacts between the receptor and the peptide backbone at the current time. Additional MLPs embed the timestep, the residue sequence, and the dihedral angles, and these embeddings are fused with the IPA output. Separate MLP heads then read the fused representation and recover the clean states. As the count of side-chain dihedral angles can by itself suggest the residue type, that cue is masked out of the type head, so the type prediction cannot take a shortcut through the torsion dimensionality. The decoder uses the same widths as the encoder: 128 for residues and 64 for pairs, and stacks 6 IPA blocks.

B.2 Training Details

Reparameterized Flow Matching Objectives. Instead of directly predict the vector field on four modalities, we predict the clean samples $\hat{x}_1$, $\hat{R}_1$, $\hat{\chi}_1$, and $\hat{s}_1$, and the vector fields are then derived from the above mentioned definitions of the conditional flows i.e. by substituting the predictions for the ground-truth endpoints in equation 19, equation 20, equation 22, and equation 24,

$$\begin{aligned} \hat{v}^{\text{pos}} &= \frac{\hat{x}_1 - x_t}{1-t}, & \hat{v}^{\text{ang}} &= \text{wrap}\left(\frac{\hat{\chi}_1 - \chi_t}{1-t}\right), \\ \hat{v}^{\text{ori}} &= \frac{\log_{R_t}(\hat{R}_1)}{1-t}, & \hat{v}^{\text{type}} &= \frac{\hat{s}_1 - s_t}{1-t}, \end{aligned} \tag{15}$$

where we omit the residue index j for brevity.

Training Algorithm. Here, we outline the training process of PepContra-base in Algorithm 1.

Algorithm 1 Training Process of PepContra-base.

Require: receptor $\mathcal{C}^{\text{rec}}$, crystal peptide $\mathcal{C}_1^{\text{pep}} = (x_1, R_1, \chi_1, a_1)$, parameters θ
1: $(h, z) \leftarrow \text{Enc}_\theta(\mathcal{C}^{\text{rec}})$
2: $s_1 \leftarrow \text{logit}(a_1)$; set $\chi_{1,j} \leftarrow 0$ on slots unused by a_1
3: $t \sim \text{Unif}[\varepsilon, 1-\varepsilon]$, $\varepsilon = 10^{-2}$
4: $x_0 \sim \mathcal{N}(0, I)$, centered on the peptide; $R_0 \sim U_{\text{SO}(3)}$; $\chi_0 \sim U(\mathbb{T}^5)$; $s_0 \sim \mathcal{N}(0, K^2 I)$
5: Form $\mathcal{C}_t^{\text{pep}}$ on peptide residues; hold the receptor at the crystal
6: $(\hat{x}_1, \hat{R}_1, \hat{\chi}_1, p_\theta) \leftarrow \text{Dec}_\theta(\mathcal{C}_t^{\text{pep}}, t, h, z)$
7: Draw $\hat{a} \sim p_\theta$; let $m(\hat{a})$ mark the torsion slots that $\hat{a}$ uses
8: Calculate the vector fields and the loss
9: Adam update of θ on $\mathcal{L}$, averaged over generated residues

Training Settings. The base model is trained from scratch on the 9701 training complexes for 80,000 iterations. For training PepContra-base, we use 8 AMD Instinct MI250X GPUs, each with 64 GB of memory, under distributed data parallelism. The batch size is 16 on each GPU, so the global batch size is 128. We use Adam with a constant learning rate of 2.5×10^{-4} and no weight decay. FLOW-HQ and FLOW-LQ are fine-tuned from this checkpoint for 20 epochs, on the 3944 high-quality complexes and the 6340 low-quality complexes respectively, with the same 8-GPU layout and per-GPU batch size and with the learning rate lowered to 1×10^{-4}. Gradients are clipped to a global norm of 100 throughout, because updates inside the IPA blocks can otherwise become large.

Influence of Training Iterations. The base model is trained for up to 240,000 iterations, and we compare the checkpoints at 80,000, 130,000, and 210,000 on the same 40 designs per target (Table 5). Although the loss keeps decreasing during training, mean DockQ rises slightly at 210,000,

the best complex RMSD, the interface energy, and sequence diversity all get worse, and peptide-internal clashes increase. We therefore use the 80,000-iteration checkpoint as the base model and as the initialization for both fine-tunes. Later checkpoints are reported only as this trend.

Table 5: Base-model checkpoint on LNR, 40 designs per target. ΔG is the mean of each target's best interface energy.

Iterations	DockQ ↑	ΔG ↓	IMP ↑	C-RMSD ↓	Clash_{in} ↓	Seq. ↑
80k	0.519	**−25.51**	20.4%	**2.37**	**0.447**	**0.684**
130k	0.520	−24.93	**21.5%**	2.44	0.571	0.589
210k	**0.540**	−24.39	20.4%	3.19	0.502	0.409

B.3 Sampling Details

Side-chain and Sequence Modeling Analysis. Full-atom co-design models such as PepFlow (Li et al., 2024) suffer from an inconsistency between training and sampling for side-chain and sequence modeling. In training, the residue type and the side-chain torsions of a residue are corrupted by the *same* time t. Since t is drawn uniformly, the pair (s_t^j, χ_t^j) sweeps one continuous trajectory from the prior to the native residue, and the network sees every type–rotamer combination at every noise level when training. We denote $\mathcal{J}(a) \subseteq \{1, \ldots, 5\}$ for the torsion slots that a type a possesses and $m(a) \in \{0, 1\}^5$ for the corresponding mask, the slots in $\mathcal{J}(a_1^j)$ are interpolated between their native values and the pre-defined initialization, whereas the remaining slots are interpolated between 0 and that same initialization, and the torsion loss are evaluated only on the former. Existence is thus decided once and for all by the native type, and the type–torsion pair stays continuous in t along the entire trajectory.

At sampling time this coupling is lost. Every Euler step re-predicts a clean sample, and the predicted type is possible to change from one step to the next, so the endpoint $\hat{\chi}_1$ is no longer a single consistent target over the time interval. For the backbone the change is harmless: every amino acid carries the same four backbone heavy atoms, so the frame updates stay continuous whatever the sequence prediction does. For the side chains it is not, because the very existence of a torsion is determined by the type. PepFlow resolves the mismatch by hard masking, forcing to zero at every step the slots that the freshly sampled type does not possess (Algorithm 2). The state carried into the next step is then a mixture of slots that have been advanced along the flow and slots that were just reset, and a position whose predicted type oscillates over a few steps has its side chain zeroed and restarted each time the type changes. We observe this discontinuity to be severe enough to dominate the sampling error even when the predicted vector field is accurate: the torsion state leaves the probability path on which the network was trained, the next prediction is therefore queried off distribution, and through the shared decoder the damage propagates back into the type head.

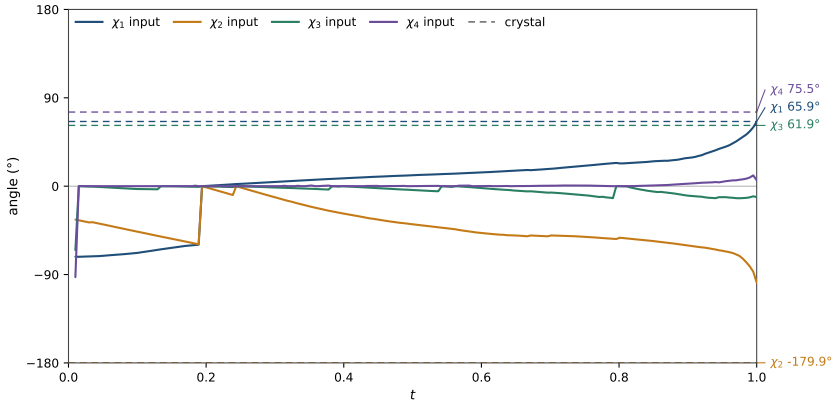


Figure 6: Side-chain rotamer trajectory during sampling. We keep recording the four torsion angle update on the last residue generated for target 1AZE.

We provide a visualization in Figure 6 as supporting evidence. In the case shown, the final designed sequence coincides with the native one, so the sampled rotamers can be compared directly against their native values. For most of the trajectory the torsion updates move in the correct direction, and only the last few steps go astray. The hard masking applied after each amino-acid type update at the end of an Euler step breaks this continuity, and when the reset happens late in sampling, the zeroed slot biases the model into treating the rotamer as absent. Because during training, a near-zero torsion input at t close to 1 occurs only for slots that the residue does not possess.

Sampling Algorithm. Here, we outline the sampling algorithm of PepContra-base in Algorithm 3 and the original sampling algorithm from PepFlow (Li et al., 2024) in Algorithm 2.

Algorithm 2 Original Sampling from PepFlow (Li et al., 2024).

Require: receptor $\mathcal{C}^{\mathrm{rec}}$, number of steps N
Ensure: peptide $\mathcal{C}^{\mathrm{pep}} = (x, R, \chi, a)$

Sample $x_0 \sim \mathcal{N}(0, I)$, $R_0 \sim U_{\mathrm{SO}(3)}$, $\chi_0 \sim U(\mathbb{T}^5)$, $s_0 \sim \mathcal{N}(0, K^2 I)$
$a_0 \sim \mathrm{softmax}(s_0)$; $(x, R, \chi, s, a) \leftarrow (x_0, R_0, \chi_0, s_0, a_0)$; $t \leftarrow 0$ ▷ χ_0 is not type-masked
for $n = 1, \ldots, N-1$ **do**
  $\Delta t \leftarrow 1/N$; predict $\hat{x}_1, \hat{R}_1, \hat{\chi}_1, p_\theta(a_1 \mid \mathcal{C}_t^{\mathrm{pep}}, \mathcal{C}^{\mathrm{rec}})$
  sample $\bar{a}_1 \sim p_\theta$; $\hat{s}_1 \leftarrow \mathrm{logit}(\bar{a}_1)$; $\hat{\chi}_1 \leftarrow \hat{\chi}_1 \odot m(\bar{a}_1)$ ▷ zero unused endpoint
  $x \leftarrow x + (\hat{x}_1 - x_0)\Delta t$; $R \leftarrow \exp_R(10\Delta t \log_R(\hat{R}_1))$
  $\chi \leftarrow \exp_\chi(\Delta t \log_\chi(\hat{\chi}_1))$ ▷ bare Δt, not $\Delta t/(1-t)$
  $s \leftarrow s + (\hat{s}_1 - s_0)\Delta t$; $a \sim \mathrm{softmax}(s)$
  $\chi \leftarrow \chi \odot m(a)$ ▷ hard-zero by noisy a_t
  $t \leftarrow t + \Delta t$
end for
Predict $\hat{\mathcal{C}}_1^{\mathrm{pep}}$ at $t = 1$, zero unused $\hat{\chi}_1$ by $\bar{a}_1$, reconstruct atoms

Algorithm 3 Synchronized Euler sampling (PepContra).

Require: receptor $\mathcal{C}^{\mathrm{rec}}$, number of steps N; under guidance, use calibrated endpoints $\bar{\mathcal{C}}_1^{\mathrm{pep}}$
Ensure: peptide $\mathcal{C}^{\mathrm{pep}} = (x, R, \chi, a)$

Initialize (x, R, χ, s, a) as in Algorithm 2; $t \leftarrow 0$; $a_{\mathrm{prev}} \leftarrow \emptyset$
for each type $b \in \{1, \ldots, S\}$ **do**
  $\chi^{(b)} \leftarrow \chi_0$; $\hat{\chi}_1^{(b)} \leftarrow \chi_0$; $t_{\mathrm{last}}^{(b)} \leftarrow 0$; $\mathrm{seen}^{(b)} \leftarrow \mathrm{false}$
end for
for $n = 1, \ldots, N-1$ **do**
  $\Delta t \leftarrow 1/N$; $t' \leftarrow t + \Delta t$
  Predict $\hat{x}_1, \hat{R}_1, \hat{\chi}_1, p_\theta(a_1 \mid \mathcal{C}_t^{\mathrm{pep}}, \mathcal{C}^{\mathrm{rec}})$; sample $\bar{a}_1 \sim p_\theta$
  $\hat{s}_1 \leftarrow \mathrm{logit}(\bar{a}_1)$; $\hat{\chi}_1 \leftarrow \hat{\chi}_1 \odot m(\bar{a}_1)$ ▷ zero unused endpoint
  Euler x, R, s as in Algorithm 2; $a \sim \mathrm{softmax}(s)$ ▷ a is embedding only
  $\chi \leftarrow \exp_\chi\big(\frac{\Delta t}{1-t} \log_\chi(\hat{\chi}_1)\big)$ ▷ flow-matched fraction
  for slot $j \notin \mathcal{J}(\bar{a}_1)$ **do**
    $\chi_j \leftarrow \exp_{\chi_{0,j}}\big(t' \log_{\chi_{0,j}}(0)\big)$ ▷ empty-slot interpolant
  end for
  if $\mathrm{seen}^{(\bar{a}_1)}$ **and** ($\bar{a}_1 \neq a_{\mathrm{prev}}$ **or** a $\chi_{1:4}$ slot re-opens) **then**
    Set $\chi_{1:4}$ by equation 14 from $\big(\chi^{(\bar{a}_1)}, \hat{\chi}_1^{(\bar{a}_1)}, t_{\mathrm{last}}^{(\bar{a}_1)}\big)$ to t'
  else
    Keep the Euler / interpolant χ ▷ first visit of this type
  end if
  Store present $\chi_{1:4}$ of $\bar{a}_1$: $\chi^{(\bar{a}_1)} \leftarrow \chi$, $\hat{\chi}_1^{(\bar{a}_1)} \leftarrow \hat{\chi}_1$, $t_{\mathrm{last}}^{(\bar{a}_1)} \leftarrow t'$; freeze the other $S-1$ types
  $\mathrm{seen}^{(\bar{a}_1)} \leftarrow \mathrm{true}$; $a_{\mathrm{prev}} \leftarrow \bar{a}_1$; $t \leftarrow t'$
end for
Predict $\hat{\mathcal{C}}_1^{\mathrm{pep}}$ at $t = 1$, zero unused $\hat{\chi}_1$ by $\bar{a}_1$, reconstruct atoms

C More Results on Peptide Co-design

Ablation Study on Designed Length. We further investigate how our proposed method influence the co-design performance of peptide in different length. In Figure 7, we split the designs by peptide length and compares the interface-energy distribution of PepContra with its two ablations and with the native peptide. The three lengths do not share one energy scale: native short peptides sit at higher ΔG, and native long peptides at lower ΔG. In every group, PepContra moves the median toward this native low-energy side. On short peptides the median is already close to the native median, so the added gain is small. The shift is clearer for long peptides. Synchronized sampling contributes to that shift on long peptides and shows little benefit on short ones. We infer that two facts may make the long-peptide case more responsive. First, the energy distribution of longer peptide is wider, so contrastive guidance has more room to concentrate probability on the lower-energy region. Second, errors also accumulate along the chain, and a correction applied at each residue, from contrastive guidance or from synchronized sampling, therefore adds up to a larger change.

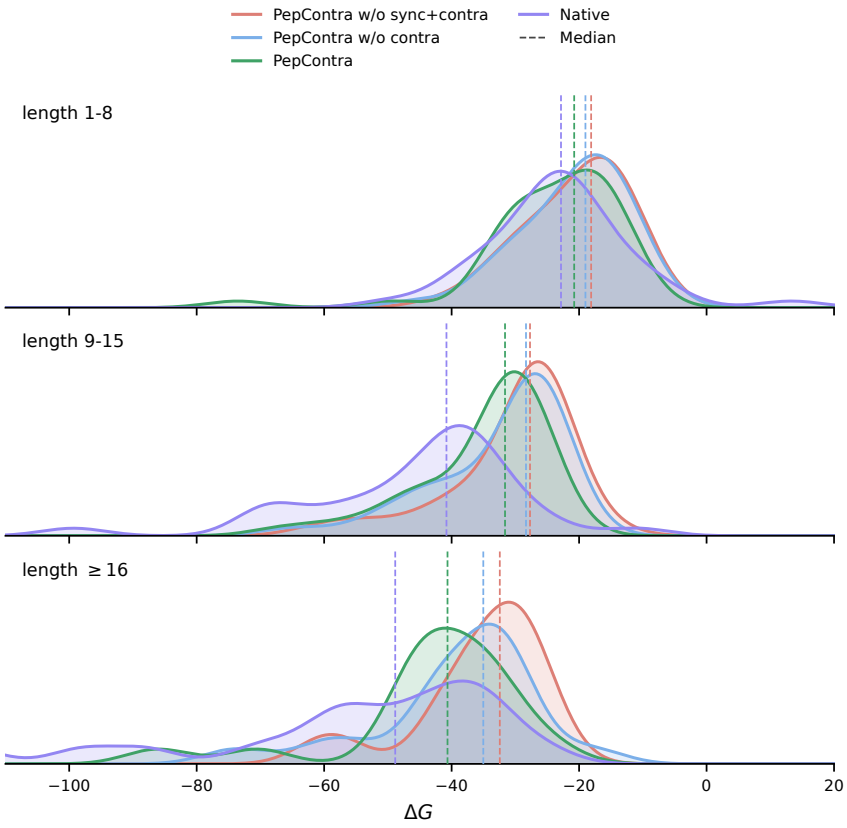


Figure 7: Interface energy on LNR for peptides length of 1–8, 9–15, and at least 16. Dashed lines mark the median of each distribution.

Energy Distribution on Testset. We compare models by a per-target summary rather than by a selecting the best case from the pool of all designs. In Figure 8, for each LNR target we take the median ΔG of its samples, then draw one box over the 93 targets. PepContra has the lowest median of these per-target medians, -15.57 REU, and the lowest interquartile range, -20.02 to -8.98 REU. The designs are therefore both lower in energy and more consistently so across targets, which is the behavior a generative peptide model needs when the samples are later ranked for virtual screening. For the same amount of candidates for screening, PepContra may have higher hit rate due to the enrichment of high-quality samples.

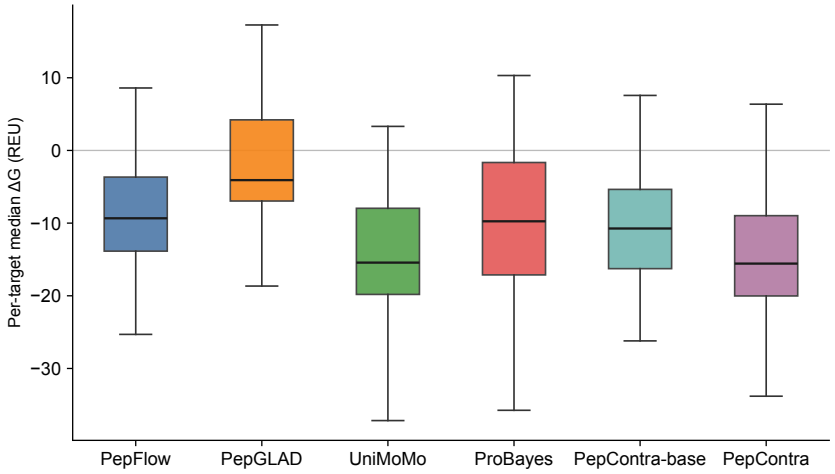


Figure 8: Per-target median interface energy on LNR. For each model, the value on a target is the median ΔG of the peptides designed for that target. The box spans the interquartile range over the 93 targets, the center line is the median of those per-target medians.

D Multi-modal Flow Matching for Peptide Design

In this section, we introduce the multi-modal flow-matching model that serves as our base generator following Li et al. (2024).

D.1 Flow Matching for Continuous and Discrete Variables

In flow matching (Lipman et al., 2023), we consider a predefined marginal probability path p_t that interpolates prior distribution p and data distribution q, i.e., $p_0 = p$ and $p_1 = q$, to train a generative model that transports a source sample $x_0 \sim p$ to a target sample $x_1 \sim q$. We use a conditional probability path $p_t(x \mid x_1)$ with $p_0(\cdot \mid x_1) \equiv p(\cdot)$ and $p_1(\cdot \mid x_1) \approx \delta(x_1)$ to construct the above p_t by $p_t(x) = \int p_t(x \mid x_1) q(x_1)\, dx_1$. The above idea of using conditional probability to construct the marginal probability path and train the generative model is utilized in both versions for modeling continuous and discrete variables.

For modeling continuous variables, given a simple implementation of $p_t(x \mid x_1)$ (e.g., Gaussian paths), we can easily derive the conditional vector field $u_t(x \mid x_1)$ that satisfies the continuity equation $\dot{p}_t(x \mid x_1) + \mathrm{div}_x[p_t(x \mid x_1) u_t(x \mid x_1)] = 0$, which means $u_t(\cdot \mid x_1)$ generates $p_t(\cdot \mid x_1)$. The marginal vector field can be defined as

$$u_t(x) \;=\; \int \frac{u_t(x \mid x_1)\, p_t(x \mid x_1)\, p(x_1)}{p_t(x)}\, dx_1. \tag{16}$$

The key insight of flow matching is that $u_t(x)$ can be proved to generate $p_t(x)$, i.e., satisfying the continuity equation $\dot{p}_t(x) + \mathrm{div}_x[p_t(x) u_t(x)] = 0$. Therefore, the conditional flow-matching (CFM) loss can be utilized to train a time-dependent neural network v_θ as follows:

$$\mathcal{L}_{\mathrm{CFM}} \;=\; \mathbb{E}_{t,\, p(x_1),\, p_t(x|x_1)} \big\| v_\theta(x,t) - u_t(x \mid x_1) \big\|^2. \tag{17}$$

We can generate samples by solving the Ordinary Differential Equation (ODE) as $\frac{dx_t}{dt} = v_\theta(x_t, t)$.

For modeling discrete variables, variants of flow matching are proposed, such as Dirichlet Flow Matching (Stark et al., 2024), Discrete Flow Matching (Campbell et al., 2024; Gat et al., 2024) and Fisher Flow Matching (Davis et al., 2024). Here we follow the line of work that lifts a discrete variable into a continuous space, so that a single ODE carries all modalities (Hoogeboom et al., 2021; Cheng et al., 2024). Let $a \in \{1, \ldots, S\}$ be a random discrete variable, e.g., one residue of a protein sequence taking on one of $S = 20$ amino acid types. Rather than designing a generative process that jumps between the S states, we embed a in a continuous space given by the geometry of the

categorical variable itself. A variable over S states corresponds to a vertex of the probability simplex $\Delta^{S-1} := \{p \in \mathbb{R}^S : p[i] \geq 0,\ \sum_{i=1}^{S} p[i] = 1\}$, and every point of Δ^{S-1} is itself a categorical distribution over those S states. The simplex therefore interpolates exactly the objects we want to generate: a trajectory across its interior is a continuously varying distribution over the S states, and the vertices it connects are the discrete values. This makes generative modeling of a discrete variable can be cast as generative modeling of a point on the simplex (Richemond et al., 2022), and the softmax supplies the map from an unconstrained vector of logits onto that simplex (Han et al., 2023).

Conditional Probability Path. We first define the conditional probability path introduced in Appendix D.1 as a product over residues and modalities,

$$p_t\big(\mathcal{C}_t^{\mathrm{pep}} \mid \mathcal{C}_1^{\mathrm{pep}}\big) \;:=\; \prod_{j=1}^{n} p_t(x_t^j \mid x_1^j)\, p_t(R_t^j \mid R_1^j)\, p_t(\chi_t^j \mid \chi_1^j)\, p_t(s_t^j \mid s_1^j), \tag{18}$$

where s^j denotes the logit representation of the type a^j defined below.

Euclidean Flow for Positions. With the isotropic Gaussian prior $x_0^j \sim \mathcal{N}(0, I_3)$, the conditional flow is the linear interpolation between the prior sample and the data point,

$$x_t^j = t\, x_1^j + (1-t)\, x_0^j, \qquad u_t^{\mathrm{pos}}\big(x_t^j \mid x_1^j, x_0^j\big) = x_1^j - x_0^j = \frac{x_1^j - x_t^j}{1-t}. \tag{19}$$

The linear interpolation favors a straight trajectory, which is the shortest path between two points in Euclidean space and contributes to the efficiency of training and sampling.

Spherical Flow for Orientations. The rotation group $\mathrm{SO}(3)$ is a smooth Riemannian manifold whose tangent space $\mathfrak{so}(3)$ consists of skew-symmetric matrices, each interpretable as an infinitesimal rotation about an axis. we establish the flow by following the corresponding geodesics, which represent paths of minimum rotational distance and interpolate orientations in a way that respects the geometry of the manifold (Chen & Lipman, 2024). From the uniform prior $R_0^j \sim \mathcal{U}(\mathrm{SO}(3))$,

$$R_t^j = \exp_{R_0^j}\!\big(t \log_{R_0^j}(R_1^j)\big), \qquad u_t^{\mathrm{ori}}\big(R_t^j \mid R_1^j\big) = \frac{\log_{R_t^j}(R_1^j)}{1-t}, \tag{20}$$

where $\exp$ and $\log$ are the exponential and logarithm maps on $\mathrm{SO}(3)$.

Toric Flow for Side-chain Torsions. A torsion angle taking values in $[0, 2\pi)$ is a point on the unit circle S^1, so the torsion vector χ^j lies on the 5-dimensional flat torus $\mathbb{T}^5 = (S^1)^5$. The flat torus can also be viewed as the quotient space $\mathbb{R}^5/(2\pi\mathbb{Z})^5$, which inherits its Riemannian metric from Euclidean space; the exponential and logarithm maps are therefore those of $\mathbb{R}^5$ up to the equivalence relation given by the periodicity of 2π, writing as

$$\mathrm{wrap}(u) := (u + \pi) \bmod 2\pi - \pi \tag{21}$$

for the map that sends a displacement to its representative of minimal length, and taking the uniform prior $\chi_0^j \sim \mathcal{U}([0, 2\pi)^5)$, the conditional flow along the geodesic and its vector field are

$$\chi_t^j = \big(\chi_0^j + t\,\mathrm{wrap}(\chi_1^j - \chi_0^j)\big) \bmod 2\pi, \qquad u_t^{\mathrm{ang}}\big(\chi_t^j \mid \chi_1^j\big) = \mathrm{wrap}\left(\frac{\chi_1^j - \chi_t^j}{1-t}\right). \tag{22}$$

Instead of the standard Euclidean distance we use the flat metric of $\mathbb{T}^5$ to compare the predicted and the ground-truth vector field, which enhances the awareness of the network of angular periodicity on the toric geometry.

Simplex Flow for Residue Types. We map a^j to a continuous space with a soft one-hot encoding operation $\mathrm{logit}(a^j) = s^j \in \mathbb{R}^S$ with a constant value $K > 0$, whose i-th entry is

$$s^j[i] = \begin{cases} K, & i = a^j, \\ -K, & \text{otherwise}, \end{cases} \tag{23}$$

so that $\mathrm{softmax}(s^j)$ is a normalized distribution with the a^j-th term close to 1 and the others close to 0, i.e. a point of Δ^{S-1} near the vertex representing a^j. We choose the prior of the logit as $s_0^j \sim \mathcal{N}(0, K^2 I)$, so that the prior on the simplex is the logistic-normal distribution by construction. The logit space is Euclidean, so the conditional flow and vector field are those of equation 19 applied to s^j,

$$s_t^j = t\, s_1^j + (1-t)\, s_0^j, \qquad u_t^{\mathrm{type}}\big(s_t^j \mid s_1^j, s_0^j\big) = s_1^j - s_0^j = \frac{s_1^j - s_t^j}{1-t}. \tag{24}$$

The linear interpolation between s_1^j and s_0^j induces a geometric-mean interpolation between the logistic-normal prior and the target categorical, so a trajectory of this flow is a continuously varying distribution over the amino-acid types.

E Implementation Details of Evaluation Metrics

Amino-acid Recovery (AAR). AAR measures how close a designed sequence is to the native binder. It is defined as the `SequenceMatcher` ratio of the two strings: the number of matching characters in the longest contiguous matching blocks, divided by the total length of the two sequences. We do not use a gapped Needleman–Wunsch identity for this metric. The reported AAR is the mean, over targets, of the best ratio among the designs for that target.

Ligand RMSD (L-RMSD) and Complex RMSD (C-RMSD). Both metrics are root-mean-square deviations of peptide C_α atoms, in angstroms, and both evaluate how close the designed backbone is to the native peptide. They differ in the frame used for the comparison. L-RMSD first superposes the designed peptide onto the native peptide by the Kabsch rotation (Kabsch, 1976) and then computes the C_α RMSD, so it measures backbone similarity after the best rigid fit and is insensitive to where the peptide sits in the pocket. C-RMSD keeps the complex in the frame obtained by aligning the pockets, and computes the peptide C_α RMSD with no further superposition, so a peptide that has the right shape but the wrong pose is still penalized. For each target we keep the lowest RMSD among its N designs. The number in the tables is the median of those per-target minima.

DockQ. DockQ (Basu & Wallner, 2016) summarizes peptide-pocket docking pose against the native complex. The DockQ program scores every native interface and returns a score in $[0, 1]$, with larger values for interfaces that recover more native contacts and have smaller interface and ligand deviations. We run it with alignment disabled, because the pocket frame is already fixed. The reported DockQ is the mean over designs of a target, then the mean over targets. This is the average interface quality of the sample, not the best design.

Interface Energy ΔG and IMP. Binding affinity is the Rosetta interface energy `dG_separated` (Alford et al., 2017) in Rosetta energy units, after a neighborhood FastRelax. The movable region is the peptide together with the pocket residues that contact it, rather than the whole receptor. Each complex is relaxed three times independently, and the three energies are averaged. Inside each relax call the default protocol repeats the minimization twice. The reported ΔG is the mean, over targets, of the lowest energy among the N designs. IMP is the fraction of targets for which that best design has a lower ΔG than the native complex. Native energies are obtained with the same three-repeat neighborhood FastRelax, and a target counts as improved when the resulting $\Delta\Delta G$ is negative.

Dihedral JSD. $\mathrm{JSD}_{\mathrm{bb}}$ and $\mathrm{JSD}_{\mathrm{sc}}$ measure whether designed backbones and side chains occupy the same torsion distribution as real peptides. Backbone dihedrals are ϕ, ψ, and ω; side-chain dihedrals are χ_1 through χ_4. Angles are histogrammed in bins of 10° on $[-180^\circ, 180^\circ)$, separately for each amino-acid type and each torsion. We then compute the Jensen–Shannon distance (Lin, 1991) between the histogram of all designed peptides and the histogram of native training peptides, and average those distances within the backbone group and within the side-chain group. The reference histogram is built from the native `peptide.pdb` files of the training pockets.

Sequence Diversity and Structural Diversity. Diversity shows how many distinct designs a model produces for one target, not how close those designs are to the native peptide. Within each

target, pairwise sequence distance is one minus a symmetric Needleman–Wunsch identity (Needleman & Wunsch, 1970). The alignment uses BLOSUM62, gap open -10, gap extend -0.5, and does not penalize end gaps. Equal-length peptides are aligned the same way. The pairwise distance is one minus this identity. Sequences are clustered by single linkage at identity 0.6. Sequence diversity is the number of clusters divided by the number of designs, averaged over targets. Structural diversity uses the same reduction on peptide C_α coordinates. The pairwise distance is the C_α RMSD in the generated frame, without a second Kabsch fit, and the single-linkage threshold is 2 Å.

Clash. Clash counts physically overlapping heavy atoms, and is reported per peptide residue so that longer peptides are not penalized only for having more atoms. Heavy atoms are the first fourteen atom slots of each residue, with same van der Waals radius 1.70 Å (C), 1.55 Å (N), 1.52 Å (O), and 1.80 Å (S), used by AlphaFold2 (Jumper et al., 2021). A pair clashes when its radius is smaller than the sum of the two radius minus a tolerance of 1.5 Å. We exclude a pair if the two atoms are the same atom, are bonded (1–2), are a bond angle apart (1–3), form the peptide-bond C–N link between consecutive residues, or are the SG atoms of two cysteines, which may be a disulfide. $\mathrm{Clash}_{\mathrm{in}}$ sums clashes among peptide atoms and divides by the number of peptide residues. $\mathrm{Clash}_{\mathrm{out}}$ does the same for clashes between peptide atoms and pocket atoms. Both numbers are pooled over all designs.

F Details of Baseline Models

Here, we introduce the details of the baseline models in experiments.

PepFlow. PepFlow (Li et al., 2024) is a target-conditioned flow-matching model for full-atom peptide co-design. Each residue is represented by a backbone frame in $\mathrm{SE}(3)$, side-chain torsions on a flat torus, and an amino-acid type on the probability simplex. The model learns a conditional vector field on each of these manifolds and integrates them jointly, so sequence, backbone, and side chains are generated together. We use the official implementation code and retrain PepFlow model with our training data. The training and sampling process all follow the original settings.

PepGLAD. PepGLAD (Kong et al., 2024) designs full-atom peptides with a geometric latent diffusion model. A variational autoencoder compresses the peptide sequence and all-atom geometry into a continuous latent code that is equivariant to rigid motions, and a diffusion model generates that code conditioned on the receptor. The designed peptide is then decoded back to a sequence and a full-atom structure. This separates the generative process from the raw coordinate space used by flow-matching co-design. PepGLAD is trained with two stage. The first stage use more than 70k synthetical data derived from protein-protein interaction to train the autoencoder to reconstruct structures. Then the model is trained on PepBench, which is a subset of our training data. PepGLAD is also tested on LNR dataset, so we directly use the checkpoint and sampling code from the official repository.

UniMoMo. UniMoMo (Kong et al., 2025b) extends the same latent-diffusion idea from peptides to a single model for several binder classes, including peptides, antibodies, and small molecules. A shared latent space and a domain-aware decoder let one network train on complexes from more than one molecular type. The checkpoint we compare against is the released unified model, which is pretrained on a much larger corpus than our peptide-only set, including antibody and small-molecule complexes (Dunbar et al., 2014; Francoeur et al., 2020). It is not that equal for our comparison because it is reported from the paper that the performance improves a lot from cross-domain training, with more than 6 REU in ΔG and 10% in IMP. We use the unified checkpoint and official sampling implementation on LNR for comparison.

ProBayes. ProBayes (Wu et al., 2025b) co-designs all-atom proteins with a Bayesian flow network (Graves et al., 2025). Instead of one joint flow over coordinates and residue types, it keeps a separate Bayesian flow for each modality and couples them with a rationalized inference step. That step is meant to stop the sequence head from reading structural shortcuts, such as the number of side-chain torsions, that would not be available in the same way at sampling time. We directly use the checkpoint and the implementation for sampling.

G HYPERPARAMETERS

Here, we introduce the hyperparameters and the searching range in experiments. Every comparison below draws 40 peptides per LNR target and uses the implementation of Appendix E.

Fine-tuning Epochs. FLOW-HQ and FLOW-LQ are fine-tuned from the 80,000-iteration checkpoint at learning rate 1×10^{-4}. For the HQ set we compare from epoch 4 and epoch 20. Epoch 20 has the lower interface energy and the lower peptide-internal clash, so FLOW-HQ is the epoch-20 checkpoint. For the low-quality set we compare epochs 8, 16, 20, 30, and 40. Extra epochs do not make the low-quality flow worse in a consistent way: ΔG stays near -23 and remains well above the base model, which is the gap contrastive guidance uses. All guidance runs use the epoch-20 low-quality checkpoint, the same epoch as FLOW-HQ.

Table 6: Fine-tuning epoch on LNR, 40 designs per target. The 80k row is the checkpoint both runs start from.

Flow	Epoch	DockQ ↑	ΔG ↓	IMP ↑	Clash_{in} ↓
Base	80k	0.519	−25.51	20.4%	0.447
FLOW-HQ	4	0.512	−26.64	24.7%	0.453
FLOW-HQ	8	**0.521**	−26.15	24.7%	0.436
FLOW-HQ	12	0.515	−26.89	**26.9%**	0.413
FLOW-HQ	16	0.514	−27.98	24.7%	0.336
FLOW-HQ	20	**0.516**	−28.61	**26.9%**	**0.283**
FLOW-LQ	8	**0.521**	−23.52	19.4%	0.886
FLOW-LQ	16	0.522	−23.97	22.6%	0.826
FLOW-LQ	20	0.527	**−22.39**	**17.2%**	**0.922**
FLOW-LQ	30	0.524	−23.82	19.4%	0.874
FLOW-LQ	40	0.534	−22.76	21.5%	0.850

Guidance Hyperparameter. Guidance uses three scales, as in Section 3.4: γ_{bb} is shared by backbone translation and rotation, γ_{seq} scales the residue-type update, and γ_χ scales the side-chain torsion update. We search them on 40 designs per target, starting from $(\gamma_{\text{bb}}, \gamma_{\text{seq}}, \gamma_\chi) = (0.10, 0.50, 0.25)$ and then raising the sequence and torsion scales (Table 7). Mean DockQ stays near 0.51 for every setting and does not select the scale. Raising γ_{seq} from 0.50 to 1.5 lowers ΔG from -28.74 to -29.48 and cuts the peptide-internal clash from 0.199 to 0.152; the intermediate value 1.25 is weaker on both. Raising γ_{bb} from 0.10 to 0.20 moves ΔG back up to -28.99, and setting it to 0 is worse still (-27.87). $\gamma_\chi = 0.40$ and 0.50 match on energy (-29.51 and -29.52); we keep 0.40, which also has the higher IMP (28.0% against 26.9%) and the lower clash (0.149 against 0.153). The locked setting is $\gamma_{\text{bb}} = 0.10$, $\gamma_{\text{seq}} = 1.5$, $\gamma_\chi = 0.40$.

Table 7: Guidance-scale search on LNR, 40 designs per target.

γ_{bb}	γ_{seq}	γ_χ	DockQ ↑	ΔG ↓	IMP ↑	Clash_{in} ↓
0.00	0.50	0.25	0.515	−27.87	28.0%	0.203
0.10	0.50	0.25	0.513	−28.74	28.0%	0.199
0.10	1.00	0.25	0.514	−29.24	25.8%	0.167
0.10	1.25	0.25	0.514	−28.87	26.9%	0.159
0.10	1.50	0.25	0.513	−29.48	26.9%	0.152
0.20	1.50	0.25	0.512	−28.99	26.9%	0.152
0.10	**1.50**	**0.40**	**0.514**	**−29.51**	**28.0%**	**0.149**
0.10	1.50	0.50	0.514	−29.52	26.9%	0.153

H MORE VISUALIZATION ON DESIGNED PEPTIDES

Figures 9 and 10 provide more visualization on designed peptide of PepContra.

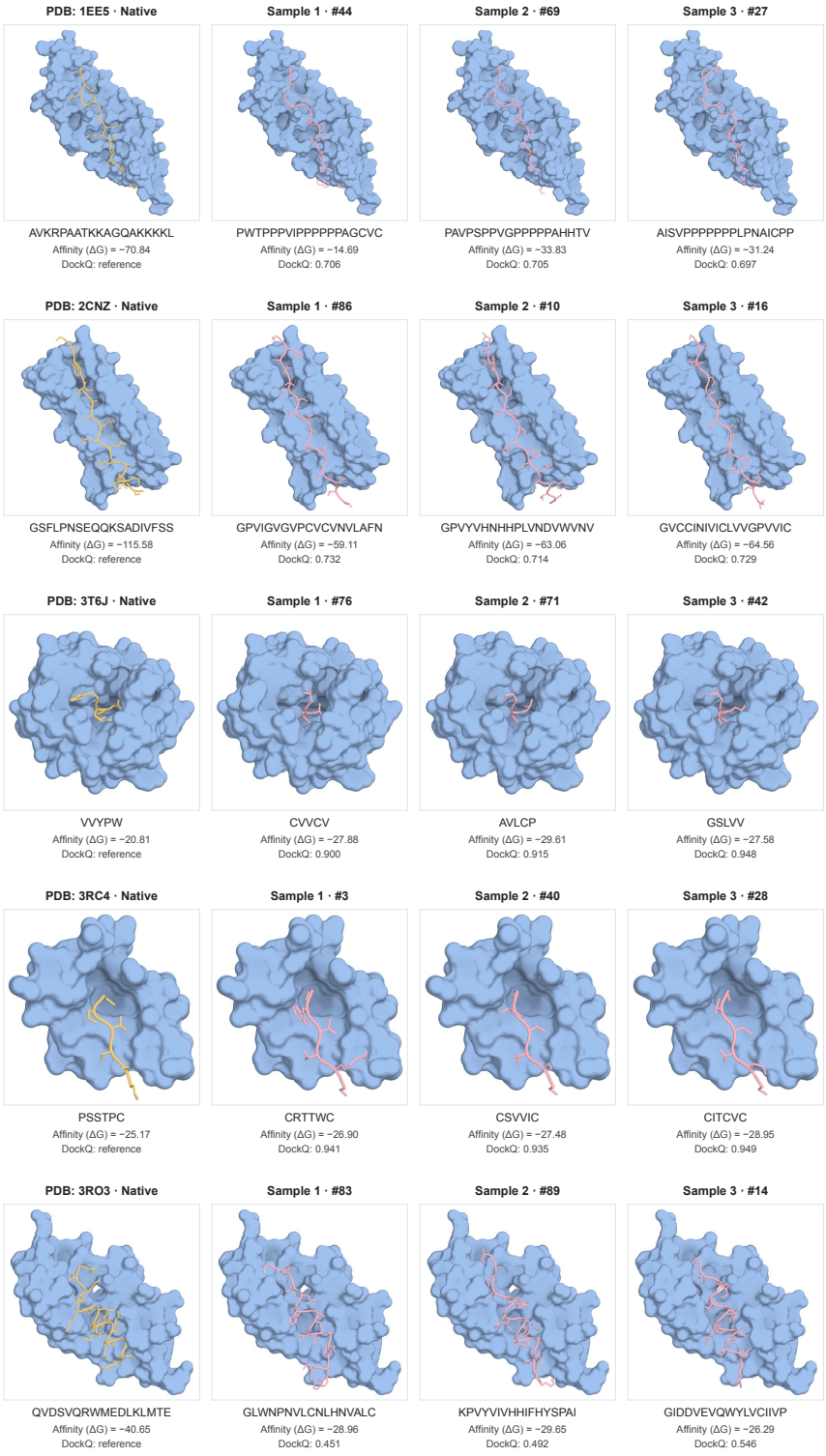


Figure 9: Designed peptides on LNR targets.

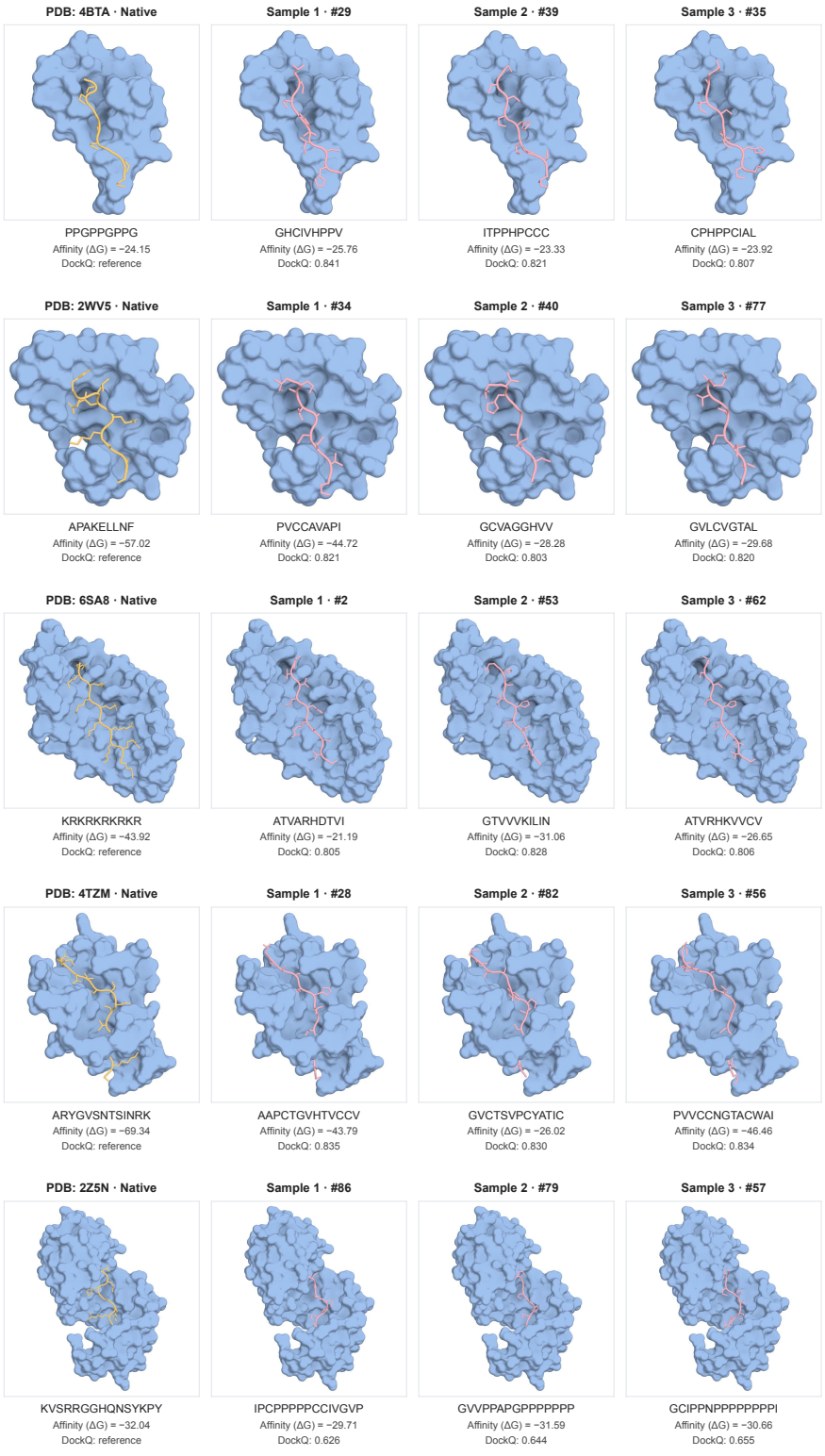


Figure 10: Designed peptides on five further LNR targets.