PEPGUIDE: INTERACTION-GUIDED LATENT DIFFUSION FOR TARGET-CONDITIONED PEPTIDE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

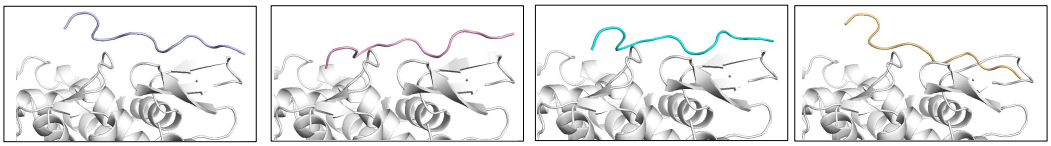
Target-conditioned peptide design poses a significant challenge in drug discovery, primarily because achieving productive peptide–target binding is subject to stringent constraints: the peptide must not only be geometrically plausible, but also establish complementary interactions with the target at the binding interface. However, existing de novo design methods struggle to satisfy these binding-critical constraints simultaneously, resulting in low binding success rates. In this paper, we propose a novel Interaction-Guided Latent Diffusion Framework (PepGuide), which explicitly injects peptide–target interaction information into the generation process to steer peptide synthesis toward conformations that bind effectively to the target. Specifically, we introduce an interaction-aware network (IANet) to capture peptide–target interactions, and integrate the resulting interaction signal into the diffusion model via a time-dependent guidance injection strategy, thereby aligning the guidance with the diffusion trajectory. In addition, we propose a geometry-aware contrastive representation learning to enhance the separability between binder and non-binder samples. Experiments on comprehensive benchmarks show that PepGuide achieves state-of-the-art performance while demonstrating strong effectiveness and robust generalization across multiple binding pockets of a therapeutically relevant target.

1 INTRODUCTION

Peptides, typically composed of 3 to 20 amino acid residues Barrett & Elmore (1998), can efficiently access cellular environments and bind specific biological targets Craik et al. (2013), making them promising biomolecules for treating complex diseases such as diabetes, obesity, hepatitis, and cancer Craik et al. (2013); Kaspar & Reichert (2013). However, discovering functional peptide binders remains challenging due to the vast peptide design space Muttenthaler et al. (2021). Traditional pipelines rely on labor-intensive screening or physics-based evaluations of large candidate libraries, making them slow, costly, and difficult to scale Zhou et al. (2025); Zhang et al. (2024). This motivates the development of efficient computational methods, especially deep learning-based generative models, for rapid peptide design with significant therapeutic potential.

Deep learning-based, target-conditioned peptide design has improved the efficiency of peptide discovery Ekambaram & Dokholyan (2026); Chen et al. (2024b), yet it still faces two major challenges. First: The Plausibility–Binding Gap. Generating chemically and geometrically plausible peptides is relatively easy, as it mainly depends on local constraints such as valid residue composition, reasonable bond geometry, and locally consistent structural patterns. In contrast, target-specific binding requires the peptide to satisfy a much narrower set of constraints, including adopting a compatible global pose and forming a target-compatible binding interface, as shown in Fig. 1. Second: The Interaction-to-Guidance Gap. Peptide–protein interactions provide crucial guidance for the design of target-specific binders, as they reflect hydrophobic/electrostatic complementarity and geometric matching at the binding interface. However, such information is highly complex, making it difficult to model explicitly and, more importantly, to translate into effective guidance during generation. Existing target-conditioned generative methods primarily condition on the overall target context, while fine-grained peptide–protein interactions are not explicitly translated into trajectory-aligned guidance during generation. Consequently, they may generate generally plausible peptide conformations but still struggle to achieve target-specific binding.

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.



(a) Energy: -31.62 (reference) (b) Energy: 25.78 (c) Energy: 14.74 (d) Energy: -13.82

Figure 1: Illustration of the Plausibility–Binding Gap. (a) Native reference complex (1eg4_P). (b–d) Complexes obtained by applying rigid-body translations and/or rotations to the peptide as a whole, thereby preserving its local structure and geometric plausibility. Energy < 0 indicates an energetically favorable binding state. These examples suggest that plausible peptide geometry alone is insufficient for binding and that target-specific constraints are required.

Overall, effective target-conditioned peptide design requires a mechanism that explicitly incorporates peptide–protein interaction information into the generative model. First, the model should comprehensively capture peptide–protein interaction patterns from both residue-level sequence correlations and structural complementarity, and transform them into guidance signals that can be used during generation. Second, the guidance signal should be compatible with the generative trajectory. For diffusion models Ho et al. (2020), generation is a constrained reverse denoising process that gradually transforms a noise prior into the data distribution. Therefore, external guidance signal should not arbitrarily overwrite the sampling trajectory or disrupt the boundary conditions of the diffusion process at either the noise or data endpoint. A more appropriate strategy is to introduce it as a controllable trajectory-level bias that applies moderate refinements to the peptide structure, gradually steering the sampling process toward conformations that enhance target binding.

Based on the above, we propose Interaction-Guided Latent Diffusion (PepGuide) as shown in Fig. 2, which directly incorporates binding-relevant peptide–protein interaction information into the generative process for all-atom peptide generation. Specifically, we first introduce an interaction-aware network (IANet) that extracts peptide–protein interaction features from complexes and projects them into a coordinate-aligned guidance signal. To ensure compatibility between the guidance signal and the diffusion trajectory, we further design a time-dependent guidance injection strategy that is incorporated into both diffusion training and inference, thereby preserving the boundary constraints of diffusion and maintaining consistency between training and inference. This strategy is applied during the middle-to-late stages of generation. Applying guidance too early may interfere with the recovery of intrinsic peptide geometry, whereas applying it too late may provide insufficient influence on the sampling trajectory. Therefore, guidance is introduced when intermediate peptide states retain moderate stochasticity and sufficient flexibility for correction, enabling more effective adjustment toward target-compatible interfaces. Additionally, we introduce geometry-aware contrastive representation learning in the variational autoencoder (VAE) Vahdat & Kautz (2020) latent space to enhance the separability of binder and non-binder representations, thereby improving target-specific diffusion-based generation. In summary, our key contributions include:

- We propose PepGuide, a target-conditioned peptide design framework that explicitly converts peptide–protein interactions into guidance signals to steer sampling toward binding-competent conformations. This helps address a key challenge in peptide generation: geometrically plausible peptides may still fail to bind their targets.
- We introduce a time-dependent guidance injection strategy that dynamically modulates the guidance strength along the denoising trajectory, preserving intrinsic peptide geometry while gradually biasing sampling toward target binding. In addition, a geometry-aware contrastive representation further unlocks the potential of this mechanism for target-specific generation.
- PepGuide achieves state-of-the-art performance on two benchmark datasets, with success rates of 83.17% on LNR and 85.58% on BioDB. Moreover, it consistently maintains a success rate above 80% across multiple binding pockets of the therapeutically relevant target Trop2, highlighting its effectiveness and strong generalization ability.

2 Related Work

Traditional peptide binder discovery relies on labor-intensive display-based screening (e.g., phage, yeast, and mRNA display) Hutchings & Sato (2024) with libraries up to 10^8–10^{12} variants Zhang (2023), yet remains limited by the vast search space and costly downstream validation Filius et al. (2025). Early computational pipelines therefore adopt template-based strategies, generating candi-

dates from known complexes or PDB-inspired heuristics and refining them via iterative mutation and docking Swanson et al. (2022); Hosseinzadeh et al. (2021). Alternatively, energy-based methods optimize hand-crafted or statistical scoring functions (e.g., Rosetta-style energies) Cao et al. (2022); Bryant & Elofsson (2023), but are computationally expensive Raveh et al. (2011) and prone to local minima—motivating learned generative models that amortize the design process.

Recent progress in protein/peptide design has been driven by deep generative models Lai et al. (2025); Zhai et al. (2025), which can be broadly grouped into sequence design, structure design, and sequence–structure co-design Wang et al. (2025). Sequence design methods focus on the inverse folding problem, predicting amino acid sequences conditioned on a fixed backbone, and have achieved strong performance in structure-constrained settings Krapp et al. (2024); Zheng et al. (2024) (e.g., ProteinMPNN Dauparas et al. (2022)). Structure design methods instead generate protein backbones directly in three-dimensional space, often leveraging diffusion-based models to enable de novo topology generation and conditional backbone design Ingraham et al. (2023); Alakhdar et al. (2024) (e.g., RFdiffusion Watson et al. (2023)). Sequence–structure co-design jointly models amino acid sequences and three-dimensional structures Luo et al. (2022); Kong et al. (2022), enabling their simultaneous and mutually consistent optimization (e.g., PepFlow Li et al. (2024b), PepGLAD Kong et al. (2024), PepHAR Li et al. (2024a)).

Beyond unconditional generation, recent studies have explored guided diffusion frameworks Chen et al. (2024a); Deng et al. (2025) that incorporate task-specific signals to steer the sampling process toward desired properties Song et al. (2025). Such guidance injects auxiliary signals Xu et al. (2019); Castillo et al. (2025) (e.g., classifiers, property predictors, or energy functions) into the denoising process Su et al. (2025), enabling conditional control without retraining Xi et al. (2025). In protein design, it has been used to enforce structural constraints, functional motifs, and global properties Liu et al. (2024); Gruver et al. (2023). However, existing approaches rely on coarse global signals and fail to explicitly model interfacial interactions, making it challenging to integrate binding-relevant cues into diffusion-based peptide generation.

3 The Proposed Method

3.1 Notation and Definitions

We represent both the target binding site and the peptide as graphs of residue blocks, denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each block $i \in \mathcal{V}$ corresponds to an amino-acid residue and is represented by $(A_i, \vec{\mathbf{X}}_i)$, where $A_i \in \mathbb{Z}^{n_i}$ denotes the atom types of the n_i atoms in block i, $\vec{\mathbf{X}}_i \in \mathbb{R}^{n_i \times 3}$ denotes their 3D coordinates, and n_i is the number of atoms in block i. The edge set $\mathcal{E}$ encodes the chemical connectivity, including both intra-block and inter-block bonds. We denote the binding-site graph and peptide graph by $\mathcal{G}^b = (\mathcal{V}^b, \mathcal{E}^b)$ and $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{E}^p)$, respectively, where $N_b = |\mathcal{V}^b|$ and $N_p = |\mathcal{V}^p|$ are the numbers of residue blocks in the binding site and peptide. We use $\mathbf{S}_{\text{ig}}$ to denote an interaction-guidance signal that steers peptide generation toward target-compatible interfaces. In this work, the binding site is defined as the set of target residues whose C_β atoms are within $10\,\text{Å}$ of any peptide residue. This threshold is chosen to retain sufficient local interaction context while avoiding an overly restrictive pocket definition. All modules in PepGuide are built upon Equivariant Transformers (ETs) Jiao et al. (2024) to preserve rotational and translational equivariance Liu et al. (2025).

3.2 Interaction-Aware Network

To steer diffusion-based peptide generation toward target-compatible binding, we introduce an Interaction-Aware Network (IANet) to extract a continuous interaction-guidance signal from peptide–protein complexes. Notably, since peptide–protein complex datasets typically lack interaction annotations, we use Rosetta binding energies E_{Rose} Alford et al. (2017) as surrogate supervision to train IANet to capture binding-relevant interfacial patterns.

Given the binding-site graph $\mathcal{G}^b$ and peptide graph $\mathcal{G}^p$, we construct a protein–peptide complex graph $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$, where $\mathcal{V}^c = \mathcal{V}^b \cup \mathcal{V}^p$ and $N_c = |\mathcal{V}^c| = N_b + N_p$. The complex-level atom types and coordinates are denoted as A^c and $\vec{\mathbf{X}}^c$, respectively, and $\mathcal{E}^c$ includes both intra-molecular edges and inter-molecular protein–peptide edges. To improve the robustness of interaction representations to small structural perturbations, we first add Gaussian noise to the complex coordinates, yielding $\tilde{\vec{\mathbf{X}}}^c$, and then process the perturbed complex with an interaction encoder implemented with stacked ET

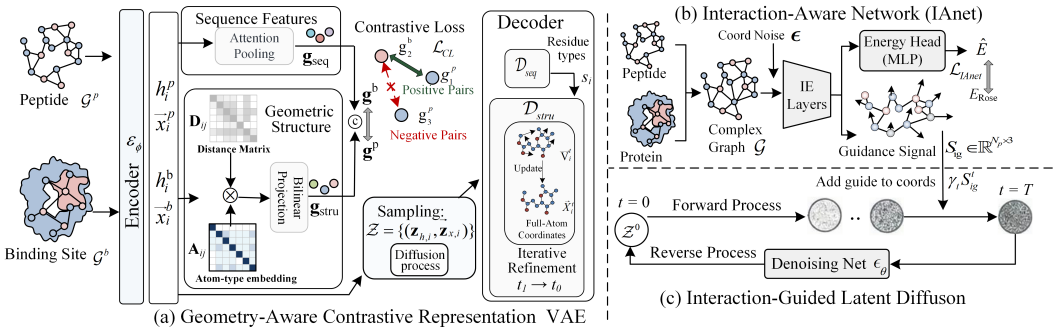


Figure 2: Overview of the proposed PepGuide framework. (a) Geometry-Aware Contrastive VAE for joint peptide sequence–structure representation learning. (b) Interaction-Aware Network (IANet) that extracts binding-relevant guidance from peptide–protein complexes. (c) Interaction-guided latent diffusion, where the guidance signal steers sampling toward binding-competent conformations.

layers:

$$\mathbf{F}_{\text{int}} = f_{\text{IE}}\left(A^c, \tilde{\vec{\mathbf{X}}}^c, \mathcal{E}^c\right), \qquad \mathbf{F}_{\text{int}} \in \mathbb{R}^{N_c \times d}, \tag{1}$$

where $f_{\text{IE}}(\cdot)$ denotes the interaction encoder, d is the feature dimension, and $\mathbf{F}_{\text{int}}$ denotes the residue-level protein–peptide interaction representation. This representation is first aggregated into a complex-level representation and then mapped to a scalar energy score:

$$\hat{E} = f_\psi(\text{Pool}(\mathbf{F}_{\text{int}})), \quad \mathcal{L}_{\text{IANet}} = \left\|\hat{E} - E_{\text{Rose}}\right\|_2^2, \tag{2}$$

where $\text{Pool}(\cdot)$ denotes the node aggregation operation, $f_\psi(\cdot)$ is an MLP and the loss is the mean squared error (MSE). More details are provided in Appendix B. Given the interaction feature $\mathbf{F}_{\text{int}} \in \mathbb{R}^{N_c \times d}$, we extract the corresponding peptide-specific feature $\mathbf{F}^p_{\text{int}} \in \mathbb{R}^{N_p \times d}$. After convergence, IANet is kept frozen during both diffusion training and inference. For notational simplicity, the overall Interaction-Aware Network is denoted as $\mathbf{F}^p_{\text{int}} = f_{\text{IANet}}(\mathcal{G}^b, \mathcal{G}^p)$.

3.3 Interaction-Guided Latent Diffusion Model

We perform diffusion in the latent space of a variational autoencoder (VAE), using $\mathcal{Z} = \{(\mathbf{z}_{h,i}, \vec{\mathbf{z}}_{x,i})\}_{i=1}^{N_p}$ as the peptide latent representation, where $\mathbf{z}_{h,i}$ and $\vec{\mathbf{z}}_{x,i}$ denote the semantic latent feature and geometric latent coordinate of the i-th residue block, respectively. Details of the VAE architecture are provided in Section 3.4.

In this section, we propose a guidance injection mechanism that incorporates interaction information extracted by IANet into the diffusion trajectory to promote target-compatible binding interfaces during sampling, while preserving the underlying diffusion dynamics and generation mechanism. First, we construct a guidance signal $\mathbf{S}_{\text{ig}}$ by projecting the interaction feature $\mathbf{F}^p_{\text{int}}$ into coordinate space: $\mathbf{S}_{\text{ig}} = f_\phi(\mathbf{F}^p_{\text{int}}), \mathbf{S}_{\text{ig}} \in \mathbb{R}^{N_p \times 3}$ where f_ϕ is an energy guidance head implemented by a multi-layer perceptron (MLP). Second, we elaborately design a time-dependent guidance injection schedule γ_t for the interaction signal $\mathbf{S}_{\text{ig}}$. The schedule mainly takes effect in the middle-to-late diffusion stages, while preserving the boundary conditions of the diffusion process and maintaining consistency between training and sampling. The γ_t can be formulated as:

$$\gamma_t = \bar{\alpha}^t \left(1 - \bar{\alpha}^t\right)^2, \qquad t \in [0, ..., T], \tag{3}$$

where $\alpha^t = 1 - \beta^t$, $\bar{\alpha}^t = \prod_{s=1}^{t}(1 - \beta^s)$, and β^t is the noise scale conforming to the cosine schedule. Our $\mathbf{S}_{\text{ig}}$ is constructed to be coordinate-aligned with the peptide geometric latent $\vec{\mathbf{z}}^p_{x,i}$. In essence, it provides each residue with a 3D offset direction pointing toward binding , which naturally encourages global pose and interface complementarity between the peptide and the protein.

We denote $\vec{u}^t_i = [\mathbf{z}^t_{h,i}, \vec{\mathbf{z}}^t_{x,i}]$ as the intermediate state of node i at time step t, and $\mathcal{Z}^t = \{\vec{u}^t_i\}_{i=1}^{N_p}$ as the corresponding peptide representation. The interaction-guidance signal is defined by $\mathbf{S}^t_{\text{ig}}$ The

forward process is then defined as:

$$q(\vec{u}_i^{\,t} \mid \vec{u}_i^{\,t-1}, \mathbf{S}_{\text{ig}}) = \mathcal{N}\Big(\mathbf{z}_{h,i}^t; \sqrt{\alpha^t}\,\mathbf{z}_{h,i}^{t-1}, \beta^t \mathrm{I}\Big) \cdot \mathcal{N}\Big(\vec{\mathbf{z}}_{x,i}^t; \sqrt{\alpha^t}\,\vec{\mathbf{z}}_{x,i}^{t-1} + \gamma_t \mathbf{S}_{\text{ig}}^t - \sqrt{\alpha^t}\gamma_{t-1}\mathbf{S}_{\text{ig}}^{t-1}, \beta^t \mathrm{I}\Big) \tag{4}$$

$$q(\vec{u}_i^{\,t} \mid \vec{u}_i^{\,0}, \mathbf{S}_{\text{ig}}) = \mathcal{N}\Big(\mathbf{z}_{h,i}^t; \sqrt{\bar{\alpha}^t}\,\mathbf{z}_{h,i}^0, (1-\bar{\alpha}^t)\mathrm{I}\Big) \cdot \mathcal{N}\Big(\vec{\mathbf{z}}_{x,i}^t; \sqrt{\bar{\alpha}^t}\vec{\mathbf{z}}_{x,i}^0 + \gamma_t \mathbf{S}_{\text{ig}}^t, (1-\bar{\alpha}^t)\mathrm{I}\Big) \tag{5}$$

Given the initial state $\vec{u}_i^{\,0}$, the state at any time t can be directly sampled as:

$$\vec{u}_i^{\,t} = \sqrt{\bar{\alpha}^t}\,\vec{u}_i^{\,0} + \sqrt{(1-\bar{\alpha}^t)}\,\epsilon_i + \gamma_t \mathbf{S}_{\text{ig}}^t, \qquad \epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

In the reverse process, since the ground-truth peptide $\mathcal{Z}^0$ is inaccessible, the corresponding guidance signal $\mathbf{S}_{\text{ig}}$ cannot be directly obtained. Therefore, we propose a reference-peptide strategy. Specifically, we first perform unguided reverse sampling without applying $\mathbf{S}_{\text{ig}}^t$ and generate K candidate peptides ($K = 3$ in this paper). These candidates are then ranked according to perplexity (PPL) scores computed by the VAE decoder, and the best-scoring peptide is selected as the reference $\mathcal{Z}^0$. The main reason is that the model has already learned a coarse geometric prior of peptides during training, and the generated reference peptide is sufficient for estimating the guidance signal $\mathbf{S}_{\text{ig}}$.The reverse process is defined as:

$$p_\theta(\vec{u}_i^{\,t-1} \mid \mathcal{Z}^{\,t}, \mathcal{G}_b) = \mathcal{N}\Big(\mathbf{z}_{h,i}^{t-1};\, \mu_{\theta,h,i}^t,\, \tilde{\beta}^t \mathrm{I}\Big) \cdot \mathcal{N}\Big(\vec{\mathbf{z}}_{x,i}^{t-1};\, \mu_{\theta,x,i}^t,\, \tilde{\beta}^t \mathrm{I}\Big), \tag{7}$$

where $\mu_{\theta,h,i}^t$ and $\mu_{\theta,x,i}^t$ denote the predicted means of the Gaussian reverse transitions for residue i in the feature space h and coordinate space x, respectively. They are written as:

$$\mu_{\theta,h,i}^t = \frac{\mathbf{z}_{h,i}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_{\theta,h}\big(\mathcal{Z}^{\,t}, \mathcal{G}_b, t\big)_i}{\sqrt{\alpha_t}},$$

$$\mu_{\theta,x,i}^t = \begin{cases} \dfrac{\vec{\mathbf{z}}_{x,i}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_{\theta,x}\big(\mathcal{Z}^{\,t}, \mathcal{G}_b, t\big)_i}{\sqrt{\alpha_t}}, & \text{for reference reverse sampling,} \\ \dfrac{\big(\vec{\mathbf{z}}_{x,i}^t - \gamma_t \mathbf{S}_{\text{ig}}^t\big) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_{\theta,x}\big(\mathcal{Z}^{\,t}, \mathcal{G}_b, t\big)_i}{\sqrt{\alpha_t}} + \gamma_{t-1}\mathbf{S}_{\text{ig}}^{t-1}, & \text{for guided reverse sampling.} \end{cases} \tag{8}$$

Here, the first branch corresponds to the unguided reverse process used to generate reference peptides, which is equivalent to setting $\gamma_t = \gamma_{t-1} = 0$. After the reference peptide is selected, $\mathbf{S}_{\text{ig}}$ can be estimated and injected into the guided reverse sampling process, as shown in the second branch. The denoising network ϵ_θ is implemented as an Equivariant Transformer (ET) to jointly predict the noise in $\mathbf{h}$ and $\mathbf{x}$. Detailed derivations are provided in Appendix C.

The training objective for the diffusion model minimizes the mean squared error (MSE) between the predicted noise and the actual noise added during the forward process which is given as:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{t\sim \mathrm{U}(1,\ldots,T)} \frac{\sum_i \big\|\epsilon_i - \epsilon_\theta\big(\mathcal{Z}^{\,t}, \mathcal{G}_b, t\big)\,[i]\big\|^2}{|\mathcal{Z}^{\,t}|}, \tag{9}$$

where $|\mathcal{Z}^{\,t}|$ indicates the number of nodes in the latent $\mathcal{Z}^{\,t}$.

3.4 Geometry-Aware Contrastive Representation Learning for VAE

As the diffusion process operates in the latent space, the quality and semantics of latent representations largely determine the generation performance. The VAE with KL regularization promotes smooth latent representations, which can attenuate binding-specific semantics: latent representations of peptides that bind to a given target may not be clearly separable from those of non-binding peptides. Consequently, diffusion-based generation has difficulty producing peptides that can specifically bind to the target. To address this, We propose a geometry-aware contrastive representation learning that explicitly integrates residue-level semantic features with atom-level geometric structure to enhance separability between binders and non-binders.

The encoder $\mathcal{E}_\phi$ takes the peptide graph $\mathcal{G}^p$ and the binding-site graph $\mathcal{G}^b$ as input, and produces contextualized residue representations:

$$\mathcal{U}^p, \mathcal{U}^b = \mathcal{E}_\phi(\mathcal{G}^p, \mathcal{G}^b), \tag{10}$$

where $\mathcal{U}^p = \{(\mathbf{h}_i^p, \vec{\mathbf{x}}_i^p)\}_{i=1}^{N_p}$, $\mathcal{U}^b = \{(\mathbf{h}_i^b, \vec{\mathbf{x}}_i^b)\}_{i=1}^{N_b}$. Here, $\mathbf{h}_i^p, \mathbf{h}_i^b \in \mathbb{R}^d$ are residue-level semantic features, and $\vec{\mathbf{x}}_i^p, \vec{\mathbf{x}}_i^b \in \mathbb{R}^3$ are the corresponding encoded coordinates.

Residue-level semantic features. For the peptide, let $\mathbf{H}^p = [\mathbf{h}_1^p, \mathbf{h}_2^p, \ldots, \mathbf{h}_{N_p}^p]^\top \in \mathbb{R}^{N_p \times d}$. We use a linear layer to produce residue-level scores $\boldsymbol{\ell}$, which are normalized by a softmax function to obtain attention weights w_i. The global sequence embedding is then computed by aggregating residue features with these attention weights, followed by layer normalization:

$$\boldsymbol{\ell} = f_\phi(\mathbf{H}^p), \qquad w_i = \frac{\exp(\ell_i)}{\sum_{j=1}^{N_p} \exp(\ell_j)}, \qquad \mathbf{g}_{\text{seq}}^p = \text{LN}_{\text{seq}}\left(\sum_{i=1}^{N_p} w_i \, \mathbf{h}_i^p\right), \tag{11}$$

where $\boldsymbol{\ell} \in \mathbb{R}^{N_p}$, $\sum_{i=1}^{N_p} w_i = 1$, and $\mathbf{g}_{\text{seq}}^p \in \mathbb{R}^d$. For the binding site, we apply the same procedure to obtain $\mathbf{g}_{\text{seq}}^b$.

Atom-level geometric structure. For a peptide residue i, we represent its all-atom structure using $C = 14$ atom channels, with residues containing fewer than 14 atoms zero-padded. Let $\mathbf{x}_{i,c}^p \in \mathbb{R}^3$ denote the coordinate of the c-th atom channel, and let $\mathbf{A}_i \in \mathbb{R}^{C \times d_a}$ denote the corresponding atom-type embedding matrix. For a residue pair (i, j), we first compute the pairwise atom-channel distance matrix $\mathbf{D}_{ij} \in \mathbb{R}^{C \times C}$, where each entry is defined as $\mathbf{D}_{ij}[c, c'] = \|\mathbf{x}_{i,c}^p - \mathbf{x}_{j,c'}^p\|_2$. A bilinear projection with atom-type embeddings is then used to obtain a compact residue-pair geometric descriptor:

$$\mathbf{R}_{ij} = \mathbf{A}_i^\top \mathbf{D}_{ij} \mathbf{A}_j \in \mathbb{R}^{d_a \times d_a}, \tag{12}$$

we vectorize each $\mathbf{R}_{ij}$, pass it through a learnable mapping f_ϕ, and average the resulting pairwise features over the edge set $\mathcal{E}^p$ to obtain the global structural representation:

$$\mathbf{g}_{\text{stru}}^p = \text{LN}_{\text{stru}}\left(\frac{1}{|\mathcal{E}^p|} \sum_{(i,j) \in \mathcal{E}^p} f_\phi(\text{vec}(\mathbf{R}_{ij}))\right), \tag{13}$$

where $\text{vec}(\cdot)$ denotes matrix vectorization and LN_{stru} denotes layer normalization. The same procedure is applied to the binding site to obtain $\mathbf{g}_{\text{stru}}^b$.

The final peptide representation concatenates the sequence and structural embeddings: $\mathbf{g}^p = [\mathbf{g}_{\text{seq}}^p; \mathbf{g}_{\text{stru}}^p]$. The binding site representation $\mathbf{g}^b$ is obtained identically. Within a mini-batch of size B, we optimize an InfoNCE contrastive loss treating matched pairs $(\mathbf{g}_m^p, \mathbf{g}_m^b)$ as positives and all cross-pairs as negatives:

$$\mathcal{L}_{\text{CL}} = \mathcal{L}_{\text{InfoNCE}}\big(\{\mathbf{g}_m^p\}_{m=1}^B, \ \{\mathbf{g}_m^b\}_{m=1}^B\big). \tag{14}$$

For each peptide residue, we construct a semantic latent variable and a geometric latent variable using the reparameterization trick. The peptide latent representation is defined as:

$$\mathcal{Z} = \{(\mathbf{z}_{h,i}, \vec{\mathbf{z}}_{x,i})\}_{i=1}^{N_p}, \tag{15}$$

where $\mathbf{z}_{h,i}$ and $\vec{\mathbf{z}}_{x,i}$ are sampled from $\mathcal{N}(h_i; \mu_i, \sigma_i)$ and $\mathcal{N}(\vec{x}_i; \vec{\mu}_i, \vec{\sigma}_i)$, respectively.

The decoder $\mathcal{D}_\phi$ reconstructs the peptide in two stages: it first predicts residue types, and then generates full-atom coordinates conditioned on the predicted sequence via iterative coordinate refinement, akin to a lightweight flow-matching process. Specifically, the residue types are predicted as:

$$\{s_i\}_{i=1}^{N_p} = \mathcal{D}_\phi\Big(\mathbf{z}_{h,1:N_p}, \mathcal{Z}, \mathcal{G}^b\Big), \tag{16}$$

where s_i denotes the predicted residue type. Then, the decoder predicts atom-wise motion vectors and updates the full-atom coordinates:

$$\{\vec{\mathrm{V}}_i^{\,t}\}_{i=1}^{N_p} = \mathcal{D}_\phi\Big(\{(\vec{\mathbf{z}}_{x,i}^t, s_i)\}_{i=1}^{N_p}, t, \mathcal{Z}, \mathcal{G}^b\Big), \qquad \vec{\mathrm{X}}_i^{\,t-\Delta t} = \vec{\mathrm{X}}_i^{\,t} + \Delta t \, \vec{\mathrm{V}}_i^{\,t}, \tag{17}$$

where $\vec{\mathrm{X}}_i^{\,t} \in \mathbb{R}^{n_i \times 3}$ denotes the full-atom coordinate matrix of residue i at iteration step t, and $\vec{\mathrm{V}}_i^{\,t} \in \mathbb{R}^{n_i \times 3}$ denotes the corresponding atom-wise motion vectors. The time variable t decreases from 1.0 to 0.0 with a step size of $\Delta t = 0.1$, resulting in a total of 10 refinement iterations. The training objective of the autoencoder consists of reconstruction losses, KL-divergence regularization, and a contrastive loss, which jointly regularize the latent space. Additional details are provided in Appendix D.

Table 1: Quantitative comparison of different peptide generation methods on the LNR and BioDB datasets. All reported metrics are measured in % except RMSD_{C_α}.

Dataset	Method	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality				
		AAR↑	RMSD_{C_α} ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓	DockQ↑	SR↑	BNR_{Ros} ↑	BNR_{MM} ↑	BNR_{Evo} ↑
LNR	PepFlow	34.01	4.1027	92.18	**56.40**	80.17	82.48	3.52	17.85	60.82	37.89	2.24	2.61	1.68
	PepGLAD	35.17	2.9728	89.76	39.65	84.47	82.10	3.63	3.98	68.25	59.32	3.68	6.20	3.80
	PepHAR	36.29	3.6343	16.66	31.47	79.79	81.08	10.37	20.08	61.64	41.81	3.93	3.48	2.83
	PepTri	34.59	2.8027	90.48	11.85	84.51	80.31	21.65	6.51	69.81	42.01	1.99	5.03	2.39
	UniMoMo	35.08	2.6672	**91.58**	22.53	92.78	93.62	**0.52**	1.46	75.15	82.35	7.22	9.70	9.29
	PepGuide	**37.23**	**2.0549**	90.96	12.72	**93.19**	**94.43**	0.97	**1.20**	**78.59**	**83.17**	**7.88**	**11.49**	**10.49**
BioDB	PepFlow	34.89	4.0202	92.97	**56.17**	83.50	80.46	4.25	15.52	59.64	31.42	0.92	4.34	3.18
	PepGLAD	34.73	3.2684	87.89	39.44	88.87	80.41	4.23	3.77	66.41	54.78	1.75	10.08	7.45
	PepHAR	31.25	2.8312	4.62	33.80	82.27	77.65	9.77	13.43	67.93	27.20	0.83	7.71	9.22
	PepTri	34.27	3.5480	89.21	13.16	89.86	77.56	24.33	5.85	63.95	39.62	0.54	9.72	6.94
	UniMoMo	34.33	2.7404	**90.93**	23.05	**96.44**	93.31	**0.59**	1.80	74.98	82.77	6.80	17.36	14.61
	PepGuide	**35.84**	**2.6904**	89.42	11.48	96.29	**93.60**	1.03	**1.49**	**75.69**	**85.58**	**7.39**	**17.43**	**14.91**

4 EXPERIMENTS

Dataset. We construct a new dataset, BioDB, by integrating the latest versions of Q-BioLip Wei et al. (2024) and PepBDB Wen et al. (2019). After removing redundant entries, we cluster complexes by receptor sequence similarity at a 40% threshold and remove all complexes whose receptors fall into the same cluster as any test-set receptor to prevent information leakage. The final dataset contains 11,058 training, 180 validation, and 130 test complexes. We compute the binding energy for each training complex using Rosetta Alford et al. (2017) as surrogate supervision for our Interaction-Aware Network. Moreover, we leverage 70k unlabeled data from protein fragments (ProtFrag) to facilitate training of the variational autoencoderKong et al. (2024). To evaluate cross-target generalization, we use the large non-redundant (LNR) dataset Tsaban et al. (2022) as an external test set, consisting of 93 expert-curated protein–peptide complexes with canonical amino acids.

4.1 SEQUENCE-STRUCTURE CO-DESIGN

Baselines. We compare PepGuide with five representative baselines for peptide sequence–structure co-design. PepFlow Li et al. (2024b) formulates peptide generation as a flow-matching process that learns continuous transformations from simple priors to realistic peptide conformations. PepGLAD Kong et al. (2024) adopts equivariant latent diffusion to generate geometrically consistent peptide structures in a compact latent space. PepHAR Li et al. (2024a) follows a hotspot-driven autoregressive strategy, where key interfacial residues are first identified and then extended into complete peptides through iterative refinement. PepTri Nguyen et al. (2026) uses tri-guided diffusion with physics-informed, evolutionary, and mutual-information guidance to improve peptide sequence–structure consistency. UniMoMo Kong et al. (2025) uses autoencoder-based geometric latent diffusion for target-conditioned 3D binder generation across small molecules, peptides, and antibodies.

Metrics. We evaluate models using a comprehensive set of metrics: **AAR**(Amino Acid Recovery) measures the fraction of generated residues matching the native peptide sequence. $\textbf{RMSD}_{C_\alpha}$ (root-mean-square deviation) is computed as the C_α RMSD between the generated and native peptides after superimposing the target proteins. **SSR** (Secondary Structure Ratio) measures the proportion of matched DSSP-assigned secondary-structure labels between generated and native peptides Kabsch & Sander (1983). **BSR** (Binding Site Rate) measures the overlap of peptide–target binding sites. Diversity is quantified by hierarchical clustering in sequence and structure space, where two samples are assigned to the same cluster if their sequence identity is $> 40\%$ or C_α RMSD is < 4 Å. **Seq** and **Struct** denote the fractions of unique sequence and structure clusters among all candidates, respectively. **DockQ** evaluates docking interfaces by combining multiple interface-level similarity measures into a score ranging from 0 to 1. $\textbf{Clash}_{\text{in}}$ and $\textbf{Clash}_{\text{out}}$ measure residue-level steric clashes within the generated peptide and across the peptide–target interface, respectively, with a clash defined as a C_α–C_α distance below 3.657 Å Ye et al. (2025). **SR** (Success Rate) and $\textbf{BNR}_{\text{Ros}}$ (Beat-Native Ratio) measure the fractions of candidates satisfying $\Delta G < 0$ and achieving lower predicted binding energies than the native complex, respectively. To complement the Rosetta-based evaluation, $\textbf{BNR}_{\text{MM}}$ and $\textbf{BNR}_{\text{Evo}}$ denote the fractions of successfully scored candidates with lower binding energies

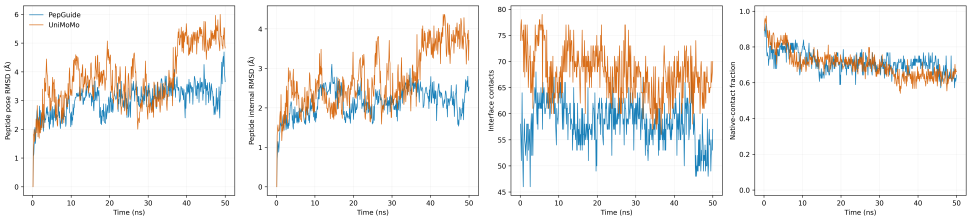


Figure 3: Molecular dynamics stability comparison of PepGuide and UniMoMo on target 2CNZ.

Table 2: Performance comparison across binding pockets.

	Pocket_1	Pocket_2	Pocket_3	Pocket_4	Pocket_5	Pocket_6
SR↑	82.5	90.0	92.5	80.0	90.0	82.5
$\Delta G \downarrow$	-50.81	-40.20	-24.70	-29.13	-34.83	-45.11

than the corresponding native complexes, evaluated using MM-GBSA Kollman et al. (2000) and EvoEF2 Huang et al. (2020), respectively.

Results. We generate 40 candidates per target for evaluation. Specifically, PepGuide first generates three reference peptides without guidance information and then produces 37 additional candidates guided by the reference-based interaction signal according to Eq. (8). Baseline methods directly generate 40 candidates for evaluation, ensuring the same candidate generation budget. To avoid potential bias from relying solely on the optimization energy function for evaluation, we further consider complementary metrics, including DockQ, BNRMM, and BNREvo. As shown in Table 1, PepGuide achieves the best overall performance on both the LNR and BioDB datasets, with particularly strong improvements in binding-related metrics. Compared with existing methods, PepGuide consistently obtains higher DockQ, SR, and BNR, indicating that it generates more accurate interfaces and more energetically favorable binders. These improvements can be attributed to the proposed interaction guidance, which explicitly incorporates peptide–protein interaction information into generation and steers sampling toward conformations with better interface compatibility.

4.2 Molecular Dynamics Analysis of Binding Stability

To further investigate the dynamic stability of the generated peptide–target complexes, we conduct all-atom molecular dynamics (MD) simulations on the 2CNZ target from the LNR dataset, comparing complexes generated by PepGuide and UniMoMo under identical simulation conditions. Detailed simulation protocols are described in Appendix E.1. The stability of the generated complexes is evaluated using four complementary metrics, including peptide pose RMSD, peptide internal RMSD, interface contacts, and native-contact fraction. Specifically, pose RMSD quantifies deviations of the peptide binding configuration relative to the initial generated complex after receptor alignment, internal RMSD measures conformational changes within the peptide after peptide alignment, interface contacts characterize the persistence of peptide–target interactions, and native-contact fraction evaluates the retention of the initially generated binding interface.

As shown in Fig. 3, the two generated complexes exhibit distinct dynamic behaviors during the simulation. PepGuide maintains lower pose and internal RMSDs throughout most of the trajectory. Its pose RMSD remains around 2.5–3.5 Å, compared with 5–6 Åfor UniMoMo in the later stage. Similarly, PepGuide maintains an internal RMSD of 1.8–2.5 Å, whereas UniMoMo reaches approximately 3–4 Å. These results indicate that the PepGuide-generated peptide undergoes less positional and conformational adjustment, maintaining a more stable binding configuration during molecular dynamics. For interface contacts, both methods maintain peptide–target interactions throughout the trajectory, while PepGuide exhibits a more stable contact pattern with smaller fluctuations and preserves a higher native-contact fraction, indicating better preservation of the initially generated binding interface. Overall, these results suggest that PepGuide generates peptide–target complexes with reduced structural relaxation and improved retention of the initial binding geometry under explicit-solvent dynamics.

4.3 Generalization Study on a New Target: Trop2.

To further assess the effectiveness and generalization of our method beyond the benchmark set, we conduct a target-specific study on human Trop2 (TACSTD2) using the experimentally determined

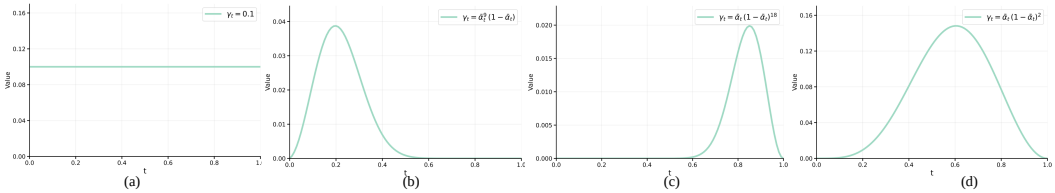


Figure 4: Illustration of different guidance schedules γ_t. (a) Constant guidance with a fixed weight. (b) Early-stage guidance, where the guidance signal mainly acts in the early denoising stage. (c) Late-stage guidance, where guidance is concentrated near the final denoising stage. (d) PepGuide's middle-to-late guidance schedule, which avoids strong guidance near the diffusion boundaries and emphasizes interaction-aware steering when the peptide state is informative and still adjustable.

Table 3: Ablation study of different guidance schedules on the BioDB dataset.

Method	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality				
	AAR↑	RMSD_{C_α} ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash_{in} (%)↓	$\text{Clash}_{\text{out}}$ (%)↓	DockQ↑	SR↑	BNR_{Ros} ↑	BNR_{MM} ↑	BNR_{Evo} ↑
No Guide	35.92	2.8249	85.50	10.91	95.11	93.12	2.41	1.21	71.76	81.78	6.97	15.87	13.64
Constant	36.82	2.8151	83.81	6.57	95.95	93.41	1.74	1.31	71.47	82.88	7.17	16.19	13.89
Early	**36.87**	3.0398	83.84	7.44	95.55	93.33	1.57	**1.06**	69.26	84.24	7.24	16.64	14.15
Late	35.62	2.9581	83.22	6.77	95.94	92.96	1.57	1.22	71.09	83.55	6.52	16.31	13.92
PepGuide	35.84	**2.6904**	**89.42**	**11.48**	**96.29**	**93.60**	**1.03**	1.49	**75.69**	**85.58**	**7.39**	**17.43**	**14.91**

structure (PDB: 7PEE) Berman et al. (2000). Trop2 is a cell-surface transmembrane glycoprotein and a key carcinoma marker and therapeutic target in many solid tumors. Goldenberg et al. (2018). We investigate six candidate binding pockets on the Trop2 surface. For each pocket, we generate 40 peptide candidates using our pocket-conditioned co-design pipeline, as shown in Appendix Fig. 5. We then compute the binding energy of each designed complex using Rosetta, and report the success rate (SR) as well as the best (lowest) binding energy observed for each pocket, as summarized in Table 2. Our designed peptides achieve an SR above 80% across all pockets, and for each pocket we obtain candidates with strong binding performance. A representative high-affinity binding conformation for Trop2 pocket 1 is shown in Appendix Fig. 6. These results indicate that our method generalizes to an unseen, clinically important target and can effectively explore multiple pockets to produce energetically favorable peptide binders.

4.4 Model Analysis

To evaluate the effect of guidance schedules, we compare different temporal weights γ_t as shown in Fig. 4, including no guidance, constant guidance, early guidance, late guidance, and the proposed middle-to-late guidance. As shown in Table 3, early guidance improves AAR and SR but leads to worse RMSD_{C_α} and DockQ, suggesting that strong guidance at noisy early stages may disturb structural recovery. Late guidance provides limited improvement because the peptide conformation has largely formed, leaving less room for interface correction. In contrast, the proposed middle-to-late schedule achieves the best overall performance, especially in RMSD_{C_α}, DockQ, SR, and BNR, indicating that interaction guidance is most effective when peptide states are sufficiently informative while still flexible for structural and interface refinement. More model analysis experiments are provided in Appendix A.

5 Conclusion

This paper presents PepGuide, an interaction-guided latent diffusion framework that explicitly incorporates binding-related interfacial constraints during generation, thereby improving the binding success rate of generated peptides. With IANet-based interaction guidance and a time-dependent injection strategy, together with geometry-aware contrastive representation learning, PepGuide steers the sampling trajectory toward more interface-complementary binding conformations while preserving global geometric validity. Experiments on the benchmark and multi-pocket Trop2 evaluation show consistently higher binding success rates. However, the limited scale and coverage of existing peptide–protein datasets may restrict the generalization of models trained on them. Future work will explore larger and more diverse peptide–protein datasets and more generalizable generative models for therapeutic target-conditioned peptide design.

AI USE STATEMENT

In this work, we used large language models (LLMs) solely for language-related assistance, including proofreading, grammar checking, and improving the clarity and readability of the manuscript.

REFERENCES

Amira Alakhdar, Barnabas Poczos, and Newell Washburn. Diffusion models in de novo drug design. *Journal of Chemical Information and Modeling*, 64(19):7238–7256, 2024.

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

Graham Campbell Barrett and Donald Trevor Elmore. *Amino acids and peptides*. Cambridge University Press, 1998.

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

Patrick Bryant and Arne Elofsson. Peptide binder design with inverse folding and protein structure prediction. *Communications Chemistry*, 6(1):229, 2023.

Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M Jude, Iva Marković, Rameshwar U Kadam, Koen HG Verschueren, et al. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, 2022.

Angela Castillo, Jonas Kohler, Juan C Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1962–1970, 2025.

Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *arXiv preprint arXiv:2404.07771*, 2024a.

Sijie Chen, Tong Lin, Ruchira Basu, Jeremy Ritchey, Shen Wang, Yichuan Luo, Xingcan Li, Dehua Pei, Levent Burak Kara, and Xiaolin Cheng. Design of target specific peptide inhibitors using generative deep learning and molecular dynamics simulations. *Nature Communications*, 15(1): 1611, 2024b. doi: 10.1038/s41467-024-45766-2.

David J Craik, David P Fairlie, Spiros Liras, and David Price. The future of peptide-based drugs. *Chemical biology & drug design*, 81(1):136–147, 2013.

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

Bowen Deng, Chang Xu, Hao Li, Yu-hao Huang, Min Hou, and Jiang Bian. Tardiff: Target-oriented diffusion guidance for synthetic electronic health record time series generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 474–485, 2025.

Srinivasan Ekambaram and Nikolay V Dokholyan. Peptide-based drug design using generative ai. *Chemical Communications*, 62(3):672–691, 2026.

Mike Filius, Thanasis Patsos, Hugo Minee, Gianluca Turco, Jingming Liu, Monika Gnatzy, Ramon SM Rooth, Andy CH Liu, Rosa DT Ta, Isa HA Rijk, et al. Generative design of high-affinity peptides using bindcraft. *bioRxiv*, pp. 2025–07, 2025.

David M Goldenberg, Rhona Stein, and Robert M Sharkey. The emergence of trophoblast cell-surface antigen 2 (trop-2) as a novel cancer target. *Oncotarget*, 9(48):28989, 2018.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

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

Parisa Hosseinzadeh, Paris R Watson, Timothy W Craven, Xinting Li, Stephen Rettie, Fátima Pardo-Avila, Asim K Bera, Vikram Khipple Mulligan, Peilong Lu, Alexander S Ford, et al. Anchor extension: a structure-guided approach to design cyclic peptides targeting enzyme active sites. *Nature Communications*, 12(1):3384, 2021.

Xiaoqiang Huang, Robin Pearce, and Yang Zhang. EvoEF2: Accurate and fast energy function for computational protein design. *Bioinformatics*, 36(4):1135–1142, 2020. doi: 10.1093/bioinformatics/btz740.

Catherine J Hutchings and Aaron K Sato. Phage display technology and its impact in the discovery of novel protein-based drugs. *Expert Opinion on Drug Discovery*, 19(8):887–915, 2024.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

Rui Jiao, Xiangzhe Kong, Ziyang Yu, Wenbing Huang, and Yang Liu. Equivariant pretrained transformer for unified geometric learning on multi-domain 3d molecules. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

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

Allan A. Kaspar and Janice M. Reichert. Future directions for peptide therapeutics development. *Drug Discovery Today*, 18(17–18):807–817, 2013. doi: 10.1016/j.drudis.2013.05.011.

Peter A. Kollman, Irina Massova, Carolina Reyes, Bernd Kuhn, Shuanghong Huo, Lillian Chong, Matthew Lee, Taisung Lee, Yong Duan, Wei Wang, Oreola Donini, Piotr Cieplak, Jaysharee Srinivasan, David A. Case, and Thomas E. Cheatham. Calculating structures and free energies of complex molecules: Combining molecular mechanics and continuum models. *Accounts of Chemical Research*, 33(12):889–897, 2000. doi: 10.1021/ar000033j.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-atom peptide design with geometric latent diffusion. *Advances in Neural Information Processing Systems*, 37:74808–74839, 2024.

Xiangzhe Kong, Zishen Zhang, Ziting Zhang, Rui Jiao, Jianzhu Ma, Wenbing Huang, Kai Liu, and Yang Liu. Unimomo: Unified generative modeling of 3d molecules for de novo binder design. *arXiv preprint arXiv:2503.19300*, 2025.

Lucien F Krapp, Fernando A Meireles, Luciano A Abriata, Jean Devillard, Sarah Vacle, Maria J Marcaida, and Matteo Dal Peraro. Context-aware geometric deep learning for protein sequence design. *Nature Communications*, 15(1):6273, 2024.

Leshan Lai, Yuansheng Liu, Bosheng Song, Keqin Li, and Xiangxiang Zeng. Deep generative models for therapeutic peptide discovery: A comprehensive review. *ACM Computing Surveys*, 57(6):1–29, 2025.

Jiahan Li, Tong Chen, Shitong Luo, Chaoran Cheng, Jiaqi Guan, Ruihan Guo, Sheng Wang, Ge Liu, Jian Peng, and Jianzhu Ma. Hotspot-driven peptide design via multi-fragment autoregressive extension. *arXiv preprint arXiv:2411.18463*, 2024a.

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. *arXiv preprint arXiv:2406.00735*, 2024b.

Wenjie Liu, Yifan Zhu, Ying Zha, Qingshan Wu, Lei Jian, and Zhihao Liu. Rotation-and permutation-equivariant quantum graph neural network for 3d graph data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yufeng Liu, Sheng Wang, Jixin Dong, Linghui Chen, Xinyu Wang, Lei Wang, Fudong Li, Chenchen Wang, Jiahai Zhang, Yuzhu Wang, et al. De novo protein design with a denoising diffusion network independent of pretrained structure prediction models. *Nature Methods*, 21(11):2107–2116, 2024.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

Markus Muttenthaler, Glenn F. King, David J. Adams, and Paul F. Alewood. Trends in peptide drug discovery. *Nature Reviews Drug Discovery*, 20(4):309–325, 2021. doi: 10.1038/s41573-020-00135-8.

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.

Barak Raveh, Nir London, Lior Zimmerman, and Ora Schueler-Furman. Rosetta flexpepdock ab-initio: simultaneous folding, docking and refinement of peptides onto their receptors. *PloS one*, 6 (4):e18934, 2011.

Mingchen Song, Xiang Deng, Zhiling Zhou, Jie Wei, Weili Guan, and Liqiang Nie. A survey on diffusion policy for robotic manipulation: Taxonomy, analysis, and future directions. *Authorea Preprints*, 2025.

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.

Chen Su, Zhengzhou Cai, Yuanhe Tian, Zhuochao Chang, Zihong Zheng, and Yan Song. Diffusion models for time series forecasting: A survey. *arXiv preprint arXiv:2507.14507*, 2025.

Sebastian Swanson, Venkatesh Sivaraman, Gevorg Grigoryan, and Amy E Keating. Tertiary motifs as building blocks for the design of protein-binding peptides. *Protein Science*, 31(6):e4322, 2022.

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.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

Dequan Wang, Zheling Tan, Jin Gao, Shaoting Zhang, Jiaqi Shen, and Yuming Lu. Ai4protein: transforming the future of protein design. *Science China Life Sciences*, pp. 1–11, 2025.

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

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.

Gongli Xi, Ye Tian, Mengyu Yang, Zhenyu Zhao, Yuchao Zhang, Xiangyang Gong, Xirong Que, and Wendong Wang. Copho: Classifier-guided conditional topology generation with persistent homology. *arXiv preprint arXiv:2512.19736*, 2025.

Bing Xu, Junfei Zhang, Rui Wang, Kun Xu, Yong-Liang Yang, Chuan Li, and Rui Tang. Adversarial monte carlo denoising with conditioned auxiliary feature modulation. *ACM Trans. Graph.*, 38(6): 224–1, 2019.

Fei Ye, Zaixiang Zheng, Dongyu Xue, Yuning Shen, Lihao Wang, Yiming Ma, Yan Wang, Xinyou Wang, Xiangxin Zhou, and Quanquan Gu. ProteinBench: A holistic evaluation of protein foundation models. In *International Conference on Learning Representations*, 2025.

Silong Zhai, Tiantao Liu, Shaolong Lin, Dan Li, Huanxiang Liu, Xiaojun Yao, and Tingjun Hou. Artificial intelligence in peptide-based drug design. *Drug Discovery Today*, pp. 104300, 2025.

Xujun Zhang, Chao Shen, Haotian Zhang, Yu Kang, Chang-Yu Hsieh, and Tingjun Hou. Advancing ligand docking through deep learning: challenges and prospects in virtual screening. *Accounts of chemical research*, 57(10):1500–1509, 2024.

Yang Zhang. Evolution of phage display libraries for therapeutic antibody discovery. In *MAbs*, volume 15, pp. 2213793. Taylor & Francis, 2023.

Yizhen Zheng, Huan Yee Koh, Maddie Yang, Li Li, Lauren T May, Geoffrey I Webb, Shirui Pan, and George Church. Large language models in drug discovery and development: From disease mechanisms to clinical trials. *arXiv preprint arXiv:2409.04481*, 2024.

Dong Zhou, Imanuel Bier, Biswajit Santra, Leif D Jacobson, Chuanjie Wu, Adiran Garaizar Suarez, Barbara Ramirez Almaguer, Haoyu Yu, Robert Abel, Richard A Friesner, et al. A robust crystal structure prediction method to support small molecule drug development with large scale validation and blind study. *Nature Communications*, 16(1):2210, 2025.

A SUPPLEMENTARY EXPERIMENTS

A.1 VISUALIZATION OF TROP2 BINDING POCKETS

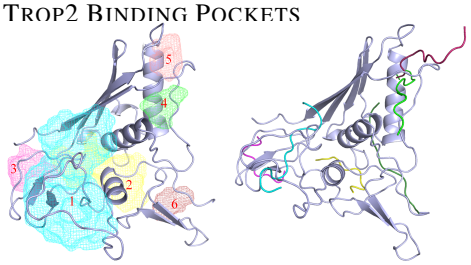


Figure 5: Visualization across pockets. Left: Different colors denote different pockets. Right: The best-binding peptide is shown for each pocket.

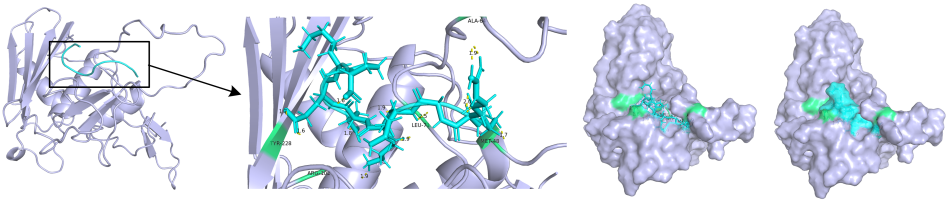


Figure 6: Example of a high-affinity binding conformation in Trop2 pocket 1 ($\Delta G = -50.81$).

A.2 ABLATION STUDY OF PEPGUIDE COMPONENTS

Our PepGuide framework consists of two key strategies:(i) geometry-aware contrastive representation learning (CRL), and (ii) interaction-aware guidance (Guide).we conduct an ablation study with the following variants: The baseline is trained without CRL and without Guide. “Baseline + CRL" adds the CRL objective to optimize peptide latent representations and enlarge the feature discrepancy between binders and non-binders. “Baseline + Guide": adds the Guide signal to steer the denoising trajectory toward geometrically compatible binding conformations with the target site. Our proposed PepGuide combines CRL and Guide to jointly enforce condition alignment and structure-/binding-aware steering.

Table 4: Ablation study of different components on the BioDB dataset.

Method	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality		
	AAR↑	RMSD_{C_α} ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash_{in} (%)↓	$\text{Clash}_{\text{out}}$ (%)↓	DockQ↑	SR↑	BNR_{Ros} ↑
Baseline	34.24	2.9545	75.95	6.58	95.72	93.47	1.45	2.26	69.27	77.09	6.41
Baseline + CRL	**35.92**	2.8249	85.50	10.91	95.11	93.12	2.41	**1.21**	71.76	81.78	6.97
Baseline + Guide	35.15	2.7442	82.56	10.60	95.32	**93.60**	1.21	1.80	72.39	81.42	6.77
PepGuide	35.84	**2.6904**	**89.42**	**11.48**	**96.29**	**93.60**	**1.03**	1.49	**75.69**	**85.58**	**7.39**

As shown in Table 4, both CRL and Guide bring clear improvements over the baseline. CRL mainly improves sequence recovery, diversity, and binding success, indicating that geometry-aware contrastive learning helps learn more discriminative latent representations. Guide further improves structural accuracy and interface quality, as reflected by lower RMSD_{C_α} and higher DockQ. Combining both components, PepGuide achieves the best overall performance across most metrics, demonstrating the complementary effects of CRL and Guide.

A.3 ABLATION STUDY OF GUIDANCE SOURCES

Table 5: Ablation study of different guidance sources on the BioDB dataset.

Method	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality		
	AAR↑	RMSD$_{C_\alpha}$ ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash$_{\text{in}}$ (%)↓	Clash$_{\text{out}}$ (%)↓	DockQ↑	SR↑	BNR$_{\text{Ros}}$ ↑
No Guide	35.92	2.8249	85.50	10.91	95.11	93.12	2.41	1.21	71.76	81.78	6.97
GT Guide	**36.34**	**2.5441**	**91.09**	9.10	94.35	**95.38**	1.15	**1.13**	**76.65**	**87.88**	**7.89**
PepGuide	35.84	2.6904	89.42	**11.48**	**96.29**	93.60	**1.03**	1.49	75.69	85.58	7.39

To investigate the effectiveness of the proposed reference-based interaction guidance, we compare three guidance settings on the BioDB dataset: (1) **No Guide**, where no interaction guidance is applied during generation; (2) **GT Guide**, where the guidance signal is computed from the native peptide–target complex using ground-truth peptide conformations; and (3) **PepGuide**, where the guidance signal is estimated from the reference peptide generated by the proposed reference-peptide strategy. As shown in Table 5, interaction guidance consistently improves binding performance over the unguided baseline. Without guidance, the model achieves an SR of 81.78% and a DockQ score of 71.76, suggesting that generative modeling alone can capture peptide plausibility but lacks explicit constraints for target-compatible interactions. Incorporating GT guidance further improves performance, reaching an SR of 87.88% and a DockQ score of 76.65, highlighting the role of accurate interaction information in directing peptides toward favorable binding interfaces. Notably, PepGuide with reference-based guidance approaches the performance of GT guidance (SR: 85.58%, DockQ: 75.69%), despite deriving guidance from generated reference peptides rather than native complexes. These results show that the proposed reference-peptide strategy effectively captures binding-relevant interaction patterns and provides practical guidance without requiring unavailable ground-truth conformations.

A.4 Effect of geometry-aware contrastive representation learning.

We study the influence of the proposed geometry-aware contrastive representation learning (CRL) by varying the contrastive weight λ_{CL}. We compare three settings, $\lambda_{\text{CL}} = 0.1$, $\lambda_{\text{CL}} = 1.0$, and the default setting $\lambda_{\text{CL}} = 0.5$, while keeping all other model components, loss weights, and training configurations unchanged.

Table 6: Ablation study of different contrastive learning weights on the BioDB dataset.

Method	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality		
	AAR↑	RMSD$_{C_\alpha}$ ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash$_{\text{in}}$ (%)↓	Clash$_{\text{out}}$ (%)↓	DockQ↑	SR↑	BNR$_{\text{Ros}}$ ↑
$\lambda_{\text{CL}} = 0.1$	36.44	2.8067	83.32	6.81	95.74	92.41	1.27	1.30	69.69	83.48	7.14
$\lambda_{\text{CL}} = 1.0$	**36.96**	2.8697	86.61	9.42	95.67	93.36	1.35	**1.22**	73.08	83.80	7.30
PepGuide (0.5)	35.84	**2.6904**	**89.42**	**11.48**	**96.29**	**93.60**	**1.03**	1.49	**75.69**	**85.58**	**7.39**

As shown in Table 6, a small contrastive weight ($\lambda_{\text{CL}} = 0.1$) provides insufficient interaction-aware supervision, leading to weaker binding quality. A larger weight ($\lambda_{\text{CL}} = 1.0$) improves AAR and Clash$_{\text{out}}$, but does not achieve the best overall performance, especially on RMSD$_{C_\alpha}$, diversity, SSR, and DockQ. In contrast, the default setting $\lambda_{\text{CL}} = 0.5$ achieves the best balance, obtaining the best results on most metrics, including RMSD$_{C_\alpha}$, diversity, SSR, BSR, Clash$_{\text{in}}$, DockQ, SR, and BNR. These results suggest that a moderate CRL weight better enhances target-compatible latent representations while preserving structural fidelity and physical validity.

A.5 Effect of candidate generation budget.

We study the influence of the candidate generation budget on BioDB. For each budget, three reference candidates are first generated and ranked by the VAE decoder PPL, with the lowest-PPL peptide selected as the reference. We then generate 7, 17, 27, or 37 additional guided candidates, resulting in total budgets of 10, 20, 30, and 40 candidates per target, respectively.

Table 7: Ablation study of different numbers of generated candidates on the BioDB dataset.

budget	Recovery		Diversity		Fidelity		Physical Validity		Binding Quality		
	AAR↑	RMSD$_{C_\alpha}$ ↓	Seq↑	Struct↑	SSR↑	BSR↑	Clash$_{\text{in}}$ (%)↓	Clash$_{\text{out}}$ (%)↓	DockQ↑	SR↑	BNR$_{\text{Ros}}$ ↑
$N = 10$	30.23	3.1832	**96.69**	**31.05**	92.04	89.01	1.07	1.58	68.55	84.59	**8.87**
$N = 20$	32.47	2.9317	93.65	18.26	94.45	91.13	**0.99**	1.63	71.97	83.08	7.14
$N = 30$	34.71	2.7513	91.28	13.58	95.70	92.54	1.01	1.49	73.91	83.48	7.54
$N = 40$ (ours)	**35.84**	**2.6904**	89.42	11.48	**96.29**	**93.60**	1.03	**1.49**	**75.69**	**85.58**	7.39

As shown in Table 7, PepGuide remains effective across different candidate generation budgets. Increasing the budget progressively improves structural fidelity, as reflected by lower RMSD$C\alpha$ and higher AAR, SSR, BSR, and DockQ. In contrast, SR remains relatively stable across budgets, while BNR$_{\text{Ros}}$ does not exhibit a monotonic trend. These results indicate that PepGuide maintains strong binding performance even under limited sampling, while additional candidates mainly improve structural and interface quality.

B Interaction-Aware Network

Given the binding-site graph $\mathcal{G}^b$ and peptide graph $\mathcal{G}^p$, we first construct a protein–peptide complex graph

$$\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c), \tag{18}$$

where the node set $\mathcal{V}^c = \mathcal{V}^b \cup \mathcal{V}^p$ consists of all residue blocks from the binding site and the peptide. Specifically,

$$\mathcal{V}^c = \left\{ \left(A_i^c, \vec{\mathbf{X}}_i^c \right) \;\middle|\; A_i^c \in \mathbb{Z}^{n_i},\ \vec{\mathbf{X}}_i^c \in \mathbb{R}^{n_i \times 3} \right\}_{i=1}^{N_c}, \tag{19}$$

where A_i^c and $\vec{\mathbf{X}}_i^c$ denote the atom types and atomic coordinates of the i-th residue block, respectively, and $N_c = |\mathcal{V}^c| = N_b + N_p$ is the total number of residue blocks in the complex. The edge set $\mathcal{E}^c$ contains both intra-molecular edges and inter-molecular protein–peptide edges.

To improve robustness to small structural perturbations, we add Gaussian noise to the input atomic coordinates:

$$\tilde{\vec{\mathbf{X}}}^c = \vec{\mathbf{X}}^c + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I). \tag{20}$$

The perturbed complex graph is then encoded by the interaction encoder:

$$\mathbf{F}_{\text{int}} = f_{\text{IE}}\left(A^c, \tilde{\vec{\mathbf{X}}}^c, \mathcal{E}^c \right), \qquad \mathbf{F}_{\text{int}} \in \mathbb{R}^{N_c \times d}, \tag{21}$$

where $f_{\text{IE}}(\cdot)$ is implemented as a stack of ET layers, and $\mathbf{F}_{\text{int}}$ denotes the residue-level interaction features for the protein–peptide complex.

Let $\mathbf{F}_{\text{int}} = [\mathbf{f}_1, \ldots, \mathbf{f}_{N_c}]^\top \in \mathbb{R}^{N_c \times d}$ denote the residue-level interaction features produced by the interaction encoder, where $\mathbf{f}_i \in \mathbb{R}^d$ is the feature of the i-th residue block. To obtain a graph-level representation for each protein–peptide complex, we adopt a multi-branch aggregation strategy consisting of attention pooling, mean pooling, and sum pooling.

For each residue block i, we first compute an unnormalized attention score

$$s_i = f_\phi(\mathbf{f}_i), \tag{22}$$

where $f_\phi(\cdot)$ is a learnable MLP that outputs a scalar. For numerical stability, the attention scores are normalized within each complex by subtracting the maximum score:

$$\tilde{s}_i = s_i - \max_{j \in \mathcal{V}^c} s_j. \tag{23}$$

The normalized attention weights are then computed as

$$\alpha_i = \frac{\exp(\tilde{s}_i)}{\sum_{j \in \mathcal{V}^c} \exp(\tilde{s}_j)}. \tag{24}$$

Based on these weights, the attention-pooled representation is

$$\mathbf{z}_{\text{att}} = \sum_{i=1}^{N_c} \alpha_i \mathbf{f}_i. \tag{25}$$

In parallel, we compute two additional graph-level summaries:

$$\mathbf{z}_{\text{mean}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{f}_i, \qquad \mathbf{z}_{\text{sum}} = \sum_{i=1}^{N_c} \mathbf{f}_i. \tag{26}$$

The three pooled features are concatenated to obtain the final complex representation, which is then used to predict the binding energy:

$$\mathbf{z}^c = [\mathbf{z}_{\text{att}} \,\|\, \mathbf{z}_{\text{mean}} \,\|\, \mathbf{z}_{\text{sum}}], \qquad \hat{E} = f_\phi(\mathbf{z}^c), \tag{27}$$

where $\|$ denotes feature concatenation. During training, we regress the sign-inverted Rosetta binding energy and optimize the mean squared error loss:

$$\tilde{E}_{\text{Rose}} = -E_{\text{Rose}}, \qquad \mathcal{L}_{\text{IANet}} = \left\| \hat{E} - \tilde{E}_{\text{Rose}} \right\|_2^2. \tag{28}$$

C Interaction-Guided Latent Diffusion Model

Notation. Let the diffusion steps be t = 1, 2, ..., T with a variance schedule $\beta_{t=1}^T$. Define

$$\alpha_t := 1 - \beta_t, \qquad \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s, \qquad \bar{\alpha}_0 := 1.$$

For each node i, we denote the latent state at step t by

$$\vec{u}_i^{\,t} = [\mathbf{z}_{h,i}^t, \vec{\mathbf{z}}_{x,i}^t],$$

where $\mathbf{z}_{h,i}^t$ is the node feature and $\vec{\mathbf{z}}_{x,i}^t$ is the coordinate. We denote the interaction-aligned guidance at step t by $\mathbf{S}_{ig}^t$ and introduce a deterministic coefficient γ_t. Define the time-dependent mean trajectory

$$m_t := \gamma_t \, \mathbf{S}_{ig}^t, \qquad m_0 = \mathbf{0} \;\; (\text{equivalently } \gamma_0 = 0).$$

C.1 Forward Process

Node feature (standard DDPM). We apply the standard DDPM forward transition to the node feature:

$$q\Big(\mathbf{z}_{h,i}^t \mid \mathbf{z}_{h,i}^{t-1}\Big) = \mathcal{N}\Big(\mathbf{z}_{h,i}^t;\ \sqrt{\alpha_t}\,\mathbf{z}_{h,i}^{t-1},\ \beta_t \mathbf{I}\Big). \tag{29}$$

Coordinate (guided DDPM). To inject guidance as a *mean shift* while preserving the standard noise evolution, we define a shifted variable:

$$\tilde{\vec{\mathbf{z}}}_{x,i}^t := \vec{\mathbf{z}}_{x,i}^t - m_t = \vec{\mathbf{z}}_{x,i}^t - \gamma_t \mathbf{S}_{ig}^t. \tag{30}$$

We then define the forward diffusion in the shifted space as a standard DDPM:

$$q\Big(\tilde{\vec{\mathbf{z}}}_{x,i}^t \mid \tilde{\vec{\mathbf{z}}}_{x,i}^{t-1}\Big) = \mathcal{N}\Big(\tilde{\vec{\mathbf{z}}}_{x,i}^t;\ \sqrt{\alpha_t}\,\tilde{\vec{\mathbf{z}}}_{x,i}^{t-1},\ \beta_t \mathbf{I}\Big). \tag{31}$$

C.1.1 Deriving the Forward Kernel in the Original Space (Eq. (4))

Coordinate one-step kernel. By substituting equation 30 into equation 31, we obtain

$$\vec{\mathbf{z}}_{x,i}^t - m_t = \sqrt{\alpha_t}\big(\vec{\mathbf{z}}_{x,i}^{t-1} - m_{t-1}\big) + \sqrt{\beta_t}\,\boldsymbol{\epsilon}_{t,i}^{(x)}, \qquad \boldsymbol{\epsilon}_{t,i}^{(x)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Rearranging terms yields the recursion in the original space:

$$\vec{\mathbf{z}}_{x,i}^t = \sqrt{\alpha_t}\,\vec{\mathbf{z}}_{x,i}^{t-1} + \big(m_t - \sqrt{\alpha_t} m_{t-1}\big) + \sqrt{\beta_t}\,\boldsymbol{\epsilon}_{t,i}^{(x)}. \tag{32}$$

Plugging $m_t = \gamma_t \mathbf{S}_{ig}^t$ into equation 32 gives

$$\vec{\mathbf{z}}_{x,i}^t = \sqrt{\alpha_t}\,\vec{\mathbf{z}}_{x,i}^{t-1} + \gamma_t \mathbf{S}_{ig}^t - \sqrt{\alpha_t}\gamma_{t-1}\mathbf{S}_{ig}^{t-1} + \sqrt{\beta_t}\,\boldsymbol{\epsilon}_{t,i}^{(x)}. \tag{33}$$

Therefore,

$$q\big(\vec{\mathbf{z}}_{x,i}^t \mid \vec{\mathbf{z}}_{x,i}^{t-1}, \mathbf{S}_{ig}\big) = \mathcal{N}\big(\vec{\mathbf{z}}_{x,i}^t;\ \sqrt{\alpha_t}\,\vec{\mathbf{z}}_{x,i}^{t-1} + \gamma_t \mathbf{S}_{ig}^t - \sqrt{\alpha_t}\gamma_{t-1}\mathbf{S}_{ig}^{t-1},\ \beta_t \mathbf{I}\big). \tag{34}$$

Joint one-step transition (Eq. (4)). Since we explicitly factorize the forward kernel between the feature and coordinate channels and inject independent Gaussian noises, combining equation 29 and equation 34 yields

$$\begin{aligned} q\big(\vec{u}_i^{\,t} \mid \vec{u}_i^{\,t-1}, \mathbf{S}_{ig}\big) &= q\Big(\mathbf{z}_{h,i}^t \mid \mathbf{z}_{h,i}^{t-1}\Big) \cdot q\big(\vec{\mathbf{z}}_{x,i}^{\,t} \mid \vec{\mathbf{z}}_{x,i}^{\,t-1}, \mathbf{S}_{ig}\big) \\ &= \mathcal{N}\Big(\mathbf{z}_{h,i}^t;\ \sqrt{\alpha_t}\,\mathbf{z}_{h,i}^{t-1},\ \beta_t \mathbf{I}\Big) \cdot \mathcal{N}\big(\vec{\mathbf{z}}_{x,i}^{\,t};\ \sqrt{\alpha_t}\,\vec{\mathbf{z}}_{x,i}^{\,t-1} + \gamma_t \mathbf{S}_{ig}^t - \sqrt{\alpha_t}\gamma_{t-1}\mathbf{S}_{ig}^{t-1},\ \beta_t \mathbf{I}\big)\,. \end{aligned} \tag{35}$$

Equation equation 35 corresponds to Eq. (4) in the main text.

C.1.2 Deriving the closed-form marginal (Eq. (5))

Node feature marginal. By standard DDPM induction,

$$q\big(\mathbf{z}_{h,i}^t \mid \mathbf{z}_{h,i}^0\big) = \mathcal{N}\big(\mathbf{z}_{h,i}^t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{z}_{h,i}^0,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,. \tag{36}$$

Coordinate marginal via shifted diffusion. From equation 31, the shifted variable follows a standard DDPM, hence

$$q\Big(\tilde{\vec{\mathbf{z}}}_{x,i}^{\,t} \mid \tilde{\vec{\mathbf{z}}}_{x,i}^{\,0}\Big) = \mathcal{N}\Big(\tilde{\vec{\mathbf{z}}}_{x,i}^{\,t};\ \sqrt{\bar{\alpha}_t}\,\tilde{\vec{\mathbf{z}}}_{x,i}^{\,0},\ (1-\bar{\alpha}_t)\mathbf{I}\Big)\,. \tag{37}$$

Using equation 30 and $m_0 = \mathbf{0}$ so that $\tilde{\vec{\mathbf{z}}}_{x,i}^{\,0} = \vec{\mathbf{z}}_{x,i}^{\,0}$, we translate back:

$$\vec{\mathbf{z}}_{x,i}^{\,t} = \tilde{\vec{\mathbf{z}}}_{x,i}^{\,t} + m_t = \tilde{\vec{\mathbf{z}}}_{x,i}^{\,t} + \gamma_t \mathbf{S}_{ig}^t.$$

Therefore,

$$q\big(\vec{\mathbf{z}}_{x,i}^{\,t} \mid \vec{\mathbf{z}}_{x,i}^{\,0}, \mathbf{S}_{ig}\big) = \mathcal{N}\big(\vec{\mathbf{z}}_{x,i}^{\,t};\ \sqrt{\bar{\alpha}_t}\,\vec{\mathbf{z}}_{x,i}^{\,0} + \gamma_t \mathbf{S}_{ig}^t,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,. \tag{38}$$

Joint marginal (Eq. equation 5). Finally, combining equation 36 and equation 38 gives

$$\begin{aligned} q\big(\vec{u}_i^{\,t} \mid \vec{u}_i^{\,0}, \mathbf{S}_{ig}\big) &= q\big(\mathbf{z}_{h,i}^t \mid \mathbf{z}_{h,i}^0\big) \cdot q\big(\vec{\mathbf{z}}_{x,i}^{\,t} \mid \vec{\mathbf{z}}_{x,i}^{\,0}, \mathbf{S}_{ig}\big) \\ &= \mathcal{N}\big(\mathbf{z}_{h,i}^t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{z}_{h,i}^0,\ (1-\bar{\alpha}_t)\mathbf{I}\big) \cdot \mathcal{N}\big(\vec{\mathbf{z}}_{x,i}^{\,t};\ \sqrt{\bar{\alpha}_t}\,\vec{\mathbf{z}}_{x,i}^{\,0} + \gamma_t \mathbf{S}_{ig}^t,\ (1-\bar{\alpha}_t)\mathbf{I}\big)\,. \end{aligned} \tag{39}$$

Equation equation 39 corresponds to Eq. equation 5 in the main text.

C.2 Reverse (denoising) process

Goal. Given the forward process in Sec. C.1, we derive the reverse-time Gaussian transition used in our method, where the denoiser predicts the forward noises for both node feature h and coordinate $\vec{x}$, and the guidance only introduces a time-dependent mean shift in the coordinate.

C.2.1 Posterior variance.

For a standard DDPM, the exact posterior $q(\cdot \mid \cdot)$ has isotropic variance

$$\tilde{\beta}_t = \beta_t\,\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}.$$

Since our guidance is a *translation* (mean shift) of the coordinate variable (see below), it does not change Gaussian covariance; therefore the same $\tilde{\beta}_t$ applies to both h and $\vec{x}$ channels:

$$\tilde{\beta}_t = \beta_t\,\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}. \tag{40}$$

C.2.2 Node feature: standard DDPM denoising mean

From the forward marginal,

$$\mathbf{z}_{h,i}^t = \sqrt{\bar{\alpha}_t}\,\mathbf{z}_{h,i}^0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_i^{(h)}, \qquad \boldsymbol{\epsilon}_i^{(h)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{41}$$

we obtain

$$\mathbf{z}_{h,i}^0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{z}_{h,i}^t - \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_i^{(h)} \right).$$

The exact DDPM posterior $q(\mathbf{z}_{h,i}^{t-1} \mid \mathbf{z}_{h,i}^t, \mathbf{z}_{h,i}^0)$ is Gaussian with variance $\tilde{\beta}_t \mathbf{I}$, and its mean can be written in the noise-parameterized form:

$$\tilde{\mu}_t^{(h)} \left(\mathbf{z}_{h,i}^t, \boldsymbol{\epsilon}_i^{(h)} \right) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_{h,i}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \boldsymbol{\epsilon}_i^{(h)} \right).$$

At test time, we replace the true noise by the denoiser prediction $\boldsymbol{\epsilon}_{\theta,h}(\mathcal{Z}^t, \mathcal{G}_b, t)_i$, yielding

$$\mu_{\theta,h,i}^t = \frac{\mathbf{z}_{h,i}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \boldsymbol{\epsilon}_{\theta,h}(\mathcal{Z}^t, \mathcal{G}_b, t)_i}{\sqrt{\alpha_t}}. \tag{42}$$

C.2.3 Coordinate: denoising mean via shifted variable

Step 1: define shifted coordinate. Introduce the shifted variable

$$\tilde{\vec{\mathbf{z}}}_{x,i}^t := \vec{\mathbf{z}}_{x,i}^t - m_t = \vec{\mathbf{z}}_{x,i}^t - \gamma_t \mathbf{S}_{ig}^t. \tag{43}$$

By construction of the forward process, $\tilde{\vec{\mathbf{z}}}_x$ follows a *standard* DDPM. Hence its marginal admits the standard form:

$$\tilde{\vec{\mathbf{z}}}_{x,i}^t = \sqrt{\bar{\alpha}_t}\, \tilde{\vec{\mathbf{z}}}_{x,i}^0 + \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_i^{(x)}, \qquad \boldsymbol{\epsilon}_i^{(x)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{44}$$

Step 2: denoising mean in shifted space. For a standard DDPM in $\tilde{\vec{\mathbf{z}}}_x$-space, the posterior $q(\tilde{\vec{\mathbf{z}}}_{x,i}^{t-1} \mid \tilde{\vec{\mathbf{z}}}_{x,i}^t, \tilde{\vec{\mathbf{z}}}_{x,i}^0)$ is Gaussian with variance $\tilde{\beta}_t \mathbf{I}$ (Eq. (11)) and mean

$$\tilde{\mu}_t^{(x)} \left(\tilde{\vec{\mathbf{z}}}_{x,i}^t, \boldsymbol{\epsilon}_i^{(x)} \right) = \frac{1}{\sqrt{\alpha_t}} \left(\tilde{\vec{\mathbf{z}}}_{x,i}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \boldsymbol{\epsilon}_i^{(x)} \right).$$

Step 3: translate back to the original coordinate. Because $\vec{\mathbf{z}}_{x,i}^{t-1} = \tilde{\vec{\mathbf{z}}}_{x,i}^{t-1} + m_{t-1}$ and $\tilde{\vec{\mathbf{z}}}_{x,i}^t = \vec{\mathbf{z}}_{x,i}^t - m_t$, the posterior mean in the original space is

$$\mu_t^{(x)} \left(\vec{\mathbf{z}}_{x,i}^t, \boldsymbol{\epsilon}_i^{(x)}, \mathbf{S}_{ig} \right) = \tilde{\mu}_t^{(x)} \left(\vec{\mathbf{z}}_{x,i}^t - m_t, \boldsymbol{\epsilon}_i^{(x)} \right) + m_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left((\vec{\mathbf{z}}_{x,i}^t - m_t) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \boldsymbol{\epsilon}_i^{(x)} \right) + m_{t-1}.$$

Substituting $m_t = \gamma_t \mathbf{S}_{ig}^t$ and $m_{t-1} = \gamma_{t-1} \mathbf{S}_{ig}^{t-1}$, and replacing the true noise by the denoiser prediction $\boldsymbol{\epsilon}_{\theta,x}(\mathcal{Z}^t, \mathcal{G}_b, t)_i$, we obtain

$$\mu_{\theta,x,i}^t = \frac{(\vec{\mathbf{z}}_{x,i}^t - \gamma_t \mathbf{S}_{ig}^t) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \boldsymbol{\epsilon}_{\theta,x}(\mathcal{Z}^t, \mathcal{G}_b, t)_i}{\sqrt{\alpha_t}} + \gamma_{t-1} \mathbf{S}_{ig}^{t-1}. \tag{45}$$

C.3 Final reverse transition used in our method.

Finally, we parameterize the reverse-time transition as a factorized Gaussian with independent isotropic covariance for each channel:

$$p_\theta \left(\vec{u}_i^{t-1} \mid \mathcal{Z}^t, \mathcal{G}_b \right) = \mathcal{N}\left(\mathbf{z}_{h,i}^{t-1};\, \mu_{\theta,h,i}^t,\, \tilde{\beta}_t \mathbf{I} \right) \cdot \mathcal{N}\left(\vec{\mathbf{z}}_{x,i}^{t-1};\, \mu_{\theta,x,i}^t,\, \tilde{\beta}_t \mathbf{I} \right). \tag{46}$$

where $\mu_{\theta,h,i}^t$ and $\mu_{\theta,x,i}^t$ are given in Eq. equation 42 and Eq. equation 45, respectively, and $\tilde{\beta}_t$ is given in Eq. (7).

D Geometry-Aware Contrastive Representation

D.1 Loss

Let Ω_p denote the set of peptide residue blocks to be generated. During training, we additionally sample a small subset of binding-site residues Ω_b^m as masked context residues. We define $\Omega = \Omega_p \cup \Omega_b^m$ and let $\mathcal{A}(\Omega)$ denote the set of atoms belonging to residue blocks in Ω. The model is trained with a combination of latent regularization, residue-type reconstruction, atom-coordinate reconstruction, interaction alignment, and topology-related losses.

Latent regularization. The encoder produces semantic and coordinate latent variables for each residue block. For semantic latents, we regularize the posterior toward a standard Gaussian prior:

$$\mathcal{L}_{\mathrm{KL}}^{h} = \frac{1}{|\Omega| d} \sum_{i \in \Omega} \mathrm{KL}\left[q_\phi(\mathbf{h}_{z,i} \mid \mathcal{G}^b, \mathcal{G}^p) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}) \right], \tag{47}$$

where d is the dimension of the semantic latent. For coordinate latents, we use an isotropic Gaussian posterior to preserve equivariance and regularize it toward a coordinate prior centered at $\vec{\mathbf{c}}_i$:

$$\mathcal{L}_{\mathrm{KL}}^{x} = \frac{1}{3|\Omega|} \sum_{i \in \Omega} \mathrm{KL}\left[q_\phi(\vec{\mathbf{x}}_{z,i} \mid \mathcal{G}^b, \mathcal{G}^p) \,\|\, \mathcal{N}(\vec{\mathbf{c}}_i, \mathbf{I}) \right]. \tag{48}$$

Residue-type reconstruction loss. Given the latent representation, the decoder predicts the residue type of each generated or masked residue block. The residue-type loss is defined as

$$\mathcal{L}_{\mathrm{type}} = -\frac{1}{|\Omega|} \sum_{i \in \Omega} \log p_\theta(s_i \mid \mathcal{G}^b, \mathcal{Z}), \tag{49}$$

where s_i denotes the ground-truth residue type of block i and $\mathcal{Z}$ denotes the latent representation.

Atom-coordinate reconstruction loss. For each atom a in residue block i, we first sample an initial coordinate from the coordinate latent prior:

$$\vec{\mathbf{x}}_{i,a}^{\,\mathrm{init}} = \vec{\mathbf{x}}_{z,i} + \boldsymbol{\epsilon}_{i,a}, \qquad \boldsymbol{\epsilon}_{i,a} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{prior}}^2 \mathbf{I}). \tag{50}$$

The target displacement from the initial state to the ground-truth coordinate is

$$\vec{\mathbf{v}}_{i,a} = \vec{\mathbf{x}}_{i,a} - \vec{\mathbf{x}}_{i,a}^{\,\mathrm{init}}. \tag{51}$$

At timestep t, the interpolated coordinate is

$$\vec{\mathbf{x}}_{i,a}^{\,t} = \vec{\mathbf{x}}_{i,a}^{\,\mathrm{init}} + (1-t)\vec{\mathbf{v}}_{i,a}. \tag{52}$$

The decoder predicts a displacement vector $\hat{\vec{\mathbf{v}}}_{i,a}^{\,t}$, and the atom-coordinate reconstruction loss is

$$\mathcal{L}_{\mathrm{coord}} = \frac{1}{|\mathcal{A}(\Omega)|} \sum_{(i,a) \in \mathcal{A}(\Omega)} \left\| \hat{\vec{\mathbf{v}}}_{i,a}^{\,t} - \vec{\mathbf{v}}_{i,a} \right\|_2^2. \tag{53}$$

Interaction-aware contrastive loss. To encourage compatibility between the binding site and the generated peptide, we construct interaction-aware representations $\mathbf{z}_k^b$ and $\mathbf{z}_k^p$ for the binding site and peptide in the k-th complex of a mini-batch. These representations are obtained by pooling residue-level features together with atom-level geometric features. For a mini-batch of size B, we compute the similarity matrix

$$\mathbf{M}_{kl} = \frac{(\mathbf{z}_k^p)^\top \mathbf{z}_l^b}{\tau}, \tag{54}$$

where τ is the temperature parameter. The symmetric contrastive loss is then defined as

$$\mathcal{L}_{\mathrm{clip}} = \frac{1}{2} \left[-\frac{1}{B} \sum_{k=1}^{B} \log \frac{\exp(\mathbf{M}_{kk})}{\sum_{l=1}^{B} \exp(\mathbf{M}_{kl})} - \frac{1}{B} \sum_{k=1}^{B} \log \frac{\exp(\mathbf{M}_{kk})}{\sum_{l=1}^{B} \exp(\mathbf{M}_{lk})} \right]. \tag{55}$$

This loss pulls matched binding-site–peptide pairs closer while pushing mismatched pairs apart in the representation space.

Local distance loss. To preserve local geometric consistency, we further constrain pairwise distances between nearby atoms. Let $\mathcal{P}_{\mathrm{loc}}$ denote the set of atom pairs whose ground-truth distance is smaller than 6 Å and where at least one atom belongs to the generated or masked region. The predicted final coordinate is computed as

$$\hat{\vec{\mathbf{x}}}_{i,a} = \vec{\mathbf{x}}_{i,a}^{\,t} + t\hat{\vec{\mathbf{v}}}_{i,a}^{\,t}. \tag{56}$$

The local distance loss is applied near the data state, i.e., when $t < t_0$ with $t_0 = 0.25$:

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{|\mathcal{P}_{\mathrm{loc}}|} \sum_{((i,a),(j,b)) \in \mathcal{P}_{\mathrm{loc}}} \rho\left(\left\| \hat{\vec{\mathbf{x}}}_{i,a} - \hat{\vec{\mathbf{x}}}_{j,b} \right\|_2 - \left\| \vec{\mathbf{x}}_{i,a} - \vec{\mathbf{x}}_{j,b} \right\|_2 \right), \tag{57}$$

where $\rho(\cdot)$ denotes the smooth-ℓ_1 loss.

Bond prediction loss. The decoder also predicts inter-block bond types for candidate atom pairs. Given the predicted bond logits $\hat{\mathbf{q}}_{uv}$ and ground-truth bond label $y_{uv} \in \{0, 1, 2, 3, 4\}$, corresponding to no bond, single, double, triple, and aromatic bonds, respectively, the bond prediction loss is

$$\mathcal{L}_{\text{bond}} = -\frac{1}{|\mathcal{E}_{\text{pred}}|} \sum_{(u,v)\in\mathcal{E}_{\text{pred}}} \log \frac{\exp(\hat{\mathbf{q}}_{uv,y_{uv}})}{\sum_{c=0}^{4} \exp(\hat{\mathbf{q}}_{uv,c})}. \tag{58}$$

Overall objective. The final training objective is a weighted sum of the above terms:

$$\mathcal{L} = \lambda_h \alpha_{\text{KL}} \mathcal{L}^h_{\text{KL}} + \lambda_x \alpha_{\text{KL}} \mathcal{L}^x_{\text{KL}} + \lambda_{\text{type}} \mathcal{L}_{\text{type}} + \lambda_{\text{coord}} \mathcal{L}_{\text{coord}} + \lambda_{\text{clip}} \mathcal{L}_{\text{clip}} + \lambda_{\text{dist}} \mathcal{L}_{\text{dist}} + \lambda_{\text{bond}} \mathcal{L}_{\text{bond}}, \tag{59}$$

where $\alpha_{\text{KL}} = \min(r_{\text{warm}}, 1)$ is a KL warm-up factor. In our implementation, the loss weights are set to $\lambda_h = 0.6$, $\lambda_x = 0.8$, $\lambda_{\text{type}} = 1.0$, $\lambda_{\text{coord}} = 1.0$, $\lambda_{\text{cl}} = 0.5$, $\lambda_{\text{dist}} = 0.5$, and $\lambda_{\text{bond}} = 0.5$. We conducted an ablation study on λ_{cl}, and set the other loss weights by referring to prior work Kong et al. (2024; 2025).

E Experiment Details

Implementation Details. Our framework is composed of three sequentially trained components: (i) a variational autoencoder (VAE) for learning residue-wise latent representations, (ii) an Interaction-Aware Network (IANet) for extracting interaction-aligned guidance signals, and (iii) an Interaction-Guided Latent Diffusion model for conditional generation in the learned latent space. All modules are optimized with AdamW using a base learning rate of 1.0×10^{-4}. We adopt a WarmupCosineLR schedule with 10 warm-up epochs, and the scheduler is updated at an epoch frequency.Training is conducted on two NVIDIA RTX 3090 GPUs. We train the VAE for up to 100 epochs, IANet for up to 80 epochs, and the Interaction-Guided Latent Diffusion model for up to 400 epochs. For all stages, we employ an early-stopping criterion based on the validation loss: training terminates if the validation loss does not improve for 10 consecutive epochs, which helps prevent overfitting and reduces unnecessary computation.

Training. In the VAE stage, we jointly train the model on the peptide dataset and ProtFrag to learn a latent space that captures the overall geometric distribution of peptides. In this stage, we do not introduce the contrastive learning (CL) loss. After the VAE converges, we freeze both the encoder and decoder, and fine-tune the intermediate latent features with a CL objective using a smaller learning rate of 3.0×10^{-5}. This two-stage strategy aims to preserve the learned peptide geometric prior while enabling the model to distinguish real binding samples from non-binding ones in the latent space.For IANet, we train it on the peptide dataset and supervise it with the binding-energy-related signal to learn interaction-aware guidance. Finally, the Interaction-Guided Latent Diffusion model is trained on the peptide dataset, leveraging the learned interaction guidance for conditional generation in the latent space.

Datasets

Table 8: Statistics of the constructed datasets.

Dataset	Split	#complexes	# clusters
BioDB	Training	11,058	1,531
BioDB	Validation	180	180
BioDB	Test	130	130
LNR	Test	93	93

Our dataset is constructed by integrating the latest versions of PepBDB and Q-BioLip. We first merge the protein–peptide complex structures from both sources and remove redundant entries, such as duplicated PDB complexes. We then refine the raw dataset with peptide-length filtering, discarding peptides that are too long (> 25 residues) or too short (< 3 residues). After these steps, we obtain a high-quality pool for downstream splitting and evaluation. To control receptor-level redundancy, we cluster receptors using MMseqs2 Steinegger & Söding (2017) with a 40% sequence-identity threshold. The final clustered dataset contains 1,531 training receptor clusters and 330 held-out

receptor clusters. To enable an unbiased generalization evaluation and prevent information leakage, complexes whose receptors belong to the same cluster are assigned to the same split. Specifically, we first select a subset of receptor clusters as the held-out set and exclude from the training data any complex whose receptor belongs to these clusters. The final split contains 11,058 training complexes and 310 held-out complexes. Among the held-out complexes, 180 are used for validation and 130 are used for testing.

ProtFrag follows the data split protocol used in PepGLAD Kong et al. (2024), and we adopt the same partition for a fair and consistent comparison. In addition, we use the large non-redundant dataset (LNR) compiled by Tsaban *et al.* as an external test set. LNR contains 93 protein–peptide complexes with canonical amino acids, curated by domain experts, and is designed to provide a high-quality, non-redundant benchmark for evaluating generalization.

Baselines. For PepGLAD, PepFlow, and PepHAR, Peptir, we re-train the models on our merged peptide dataset based on their official implementations, while keeping the hyperparameters as the default settings in the corresponding codebases. Since UniMoMo only releases the all-domain checkpoint rather than a peptide-specific one, we directly compare with its released all-domain version, which is trained on approximately 184K multi-domain molecular samples. In contrast, PepGuide is trained using only 11,058 peptide–protein complexes, further highlighting the effectiveness of our method under a much smaller training-data scale.

E.1 Molecular Dynamics Simulation Setup and Metrics

Setup. For each method, we select the generated complex with the lowest Rosetta binding energy for MD simulation. All simulations are conducted using OpenMM with the AMBER ff14SB force field and the TIP3P water model. Each complex is solvated in a periodic dodecahedral box with a minimum solute–box distance of 1.2 nm and an ionic strength of 0.15 M NaCl. Long-range electrostatic interactions are treated using the particle-mesh Ewald (PME) method with a 1.0 nm nonbonded cutoff. Bonds involving hydrogen atoms are constrained, allowing an integration time step of 2 fs. Each system is energy-minimized for up to 5,000 iterations and gradually heated to 298.15 K over 200 ps. The system is then equilibrated for 500 ps under the NVT ensemble, followed by 500 ps under the NPT ensemble at 1 atm with gradually released backbone restraints. Finally, a 50-ns unrestrained production simulation is performed.

Molecular Dynamics Metrics. We characterize the dynamic stability of the generated peptide–target complexes using four complementary metrics. **Peptide pose RMSD** measures the deviation of the peptide from its initially generated binding configuration after aligning the receptor, and therefore reflects changes in peptide position, orientation, and overall binding pose relative to the target; lower and more stable values indicate better preservation of the generated binding geometry. **Peptide internal RMSD** is computed after independently aligning the peptide and mainly captures conformational changes within the peptide itself, allowing rigid-body motion to be distinguished from internal structural rearrangement; lower values indicate better preservation of the peptide conformation. **Interface contacts** quantify the number of direct peptide–target contacts maintained during the simulation and reflect the persistence of physical association at the binding interface; a relatively stable contact number suggests sustained peptide–target interactions, whereas a pronounced decrease indicates interface weakening or disruption. **Native-contact fraction** measures the proportion of contacts from the initially generated interface that remain preserved over time, with values closer to 1 indicating stronger retention of the original interaction pattern.

Limitations

Although PepGuide achieves strong performance on the evaluated benchmarks, several limitations remain. First, the limited scale and coverage of existing peptide–protein complex datasets may leave some target families underrepresented, suggesting the need for broader future validation. Second, the current evaluation is mainly based on computational metrics and predicted binding energies, while wet-lab experimental validation is still required before practical applications. Finally, generating and evaluating multiple candidates may introduce additional computational cost during inference.

F Implementation Details

F.1 Hyperparameter Settings

Table 9 summarizes the main hyperparameters of PepGuide. We report the settings for the geometry-aware contrastive VAE, the interaction-aware network, and the interaction-guided latent diffusion model, including loss weights, latent dimensions, network architectures, and diffusion steps.

Table 9: Hyperparameters of PepGuide in sequence-structure codesign and binding conformation generation.

Name	Value	Description
Geometry-Aware Contrastive Representation Learning for VAE		
w_{KL}^{h}	0.6	Weight of KL divergence for the h latent variable.
w_{KL}^{z}	0.8	Weight of KL divergence for the z latent variable.
λ_{coord}	1.0	Weight of atom-coordinate reconstruction loss.
λ_{type}	1.0	Weight of residue-type prediction loss.
$\lambda_{\mathrm{local_dist}}$	0.5	Weight of local distance loss.
λ_{bond}	0.5	Weight of bond constraint loss.
λ_{CL}	0.5	Weight of geometry-aware contrastive loss.
embed size	512	Size of embeddings for residue types.
hidden size	512	Size of hidden layers.
h	8	Size of the latent variable for residue types in the sequences.
layers	6	Number of layers.
Interaction-Aware Network		
hidden size	512	Size of hidden layers.
layers	5	Number of layers.
Interaction-Guided Latent Diffusion Model		
hidden size	512	Size of hidden layers.
layers	6	Number of layers.
steps	200	Number of diffusion steps.

F.2 Algorithm.

To provide a clearer description of the implementation procedure, we summarize the training and inference algorithms of PepGuide in this section. The training process is divided into three stages: geometry-aware contrastive VAE training, Interaction-Aware Network (IANet) training, and interaction-guided latent diffusion training. During inference, PepGuide first generates reference peptide candidates through unguided sampling and selects the best one according to the decoder likelihood/PPL score. The selected reference peptide is then used to estimate the interaction-guidance signal, which is injected into the reverse diffusion process for final guided peptide generation.

Algorithm 1: Training Algorithm of the Geometry-Aware Contrastive VAE

Input : training complexes $\mathcal{S} = \{(\mathcal{G}^b, \mathcal{G}^p)\}$
Output : encoder $\mathcal{E}_\phi$, decoder $\mathcal{D}_\phi$

Function `Encode` $(\mathcal{E}_\phi, \mathcal{G}^b, \mathcal{G}^p)$ **:**
  $\{\boldsymbol{\mu}_i^h, \boldsymbol{\sigma}_i^h, \vec{\boldsymbol{\mu}}_i^x, \vec{\boldsymbol{\sigma}}_i^x\} \leftarrow \mathcal{E}_\phi(\mathcal{G}^b, \mathcal{G}^p)$; `// Encoding`
  $\epsilon_i^h, \epsilon_i^x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; `// Reparameterization`
  $\mathcal{Z} \leftarrow \{(\boldsymbol{\mu}_i^h + \epsilon_i^h \odot \boldsymbol{\sigma}_i^h, \vec{\boldsymbol{\mu}}_i^x + \epsilon_i^x \odot \vec{\boldsymbol{\sigma}}_i^x)\}$
  return $\mathcal{Z}$
Initialize $\mathcal{E}_\phi$ and $\mathcal{D}_\phi$;
while ϕ *has not converged* **do**
  Sample $(\mathcal{G}^b, \mathcal{G}^p) \sim \mathcal{S}$;
  $\mathcal{Z} \leftarrow \texttt{Encode}(\mathcal{E}_\phi, \mathcal{G}^b, \mathcal{G}^p)$; `// Latent sampling`
  $\hat{\mathcal{G}}^p \leftarrow \mathcal{D}_\phi(\mathcal{Z}, \mathcal{G}^b)$; `// Sequence-structure reconstruction`
  $\mathbf{g}^p, \mathbf{g}^b \leftarrow \mathrm{GeoRep}(\mathcal{G}^p, \mathcal{G}^b)$; `// Geometry-aware representations`
  $\mathcal{L}_{\mathrm{CL}} \leftarrow \mathcal{L}_{\mathrm{InfoNCE}}(\mathbf{g}^p, \mathbf{g}^b)$; `// Contrastive loss`
  $\mathcal{L}_{\mathrm{VAE}} \leftarrow \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{cl}}\mathcal{L}_{\mathrm{CL}}$; `// Calculate loss`
  $\phi \leftarrow \mathrm{optimizer}(\mathcal{L}_{\mathrm{VAE}}; \phi)$;
return $\mathcal{E}_\phi, \mathcal{D}_\phi$;

Algorithm 2: Training Algorithm of the Interaction-Aware Network

Input : training complexes $\mathcal{S} = \{(\mathcal{G}^b, \mathcal{G}^p, E_{\mathrm{Rose}})\}$
Output : Interaction-Aware Network f_{IANet}

Initialize f_{IANet};
while f_{IANet} *has not converged* **do**
  Sample $(\mathcal{G}^b, \mathcal{G}^p, E_{\mathrm{Rose}}) \sim \mathcal{S}$;
  $\mathcal{G}^c \leftarrow \mathrm{BuildComplex}(\mathcal{G}^b, \mathcal{G}^p)$; `// Construct complex graph`
  $\tilde{\vec{\mathbf{X}}}^c \leftarrow \vec{\mathbf{X}}^c + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$; `// Coordinate perturbation`
  $\mathbf{F}_{\mathrm{int}} \leftarrow f_{\mathrm{IE}}(A^c, \tilde{\vec{\mathbf{X}}}^c, \mathcal{E}^c)$; `// Interaction encoding`
  $\hat{E} \leftarrow f_\psi(\mathrm{Pool}(\mathbf{F}_{\mathrm{int}}))$; `// Energy prediction`
  $\mathcal{L}_{\mathrm{IANet}} \leftarrow \|\hat{E} - E_{\mathrm{Rose}}\|_2^2$; `// Calculate loss`
return f_{IANet};

Algorithm 3: Training Algorithm of the Interaction-Guided Latent Diffusion Model

Input : training complexes $\mathcal{S} = \{(\mathcal{G}^b, \mathcal{G}^p)\}$, trained $\mathcal{E}_\phi$, trained f_{IANet}
Output : denoising network ϵ_θ

Freeze $\mathcal{E}_\phi$ and f_{IANet};
Initialize ϵ_θ;
while θ *has not converged* **do**
  Sample $(\mathcal{G}^b, \mathcal{G}^p) \sim \mathcal{S}$;
  $\mathcal{Z}^0 \leftarrow \mathcal{E}_\phi(\mathcal{G}^b, \mathcal{G}^p)$; `// Encode peptide into latent space`
  $\mathbf{S}_{\mathrm{ig}} \leftarrow f_\phi(f_{\mathrm{IANet}}(\mathcal{G}^b, \mathcal{G}^p))$; `// Interaction guidance`
  Sample $t \sim \mathcal{U}(1, \ldots, T)$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  $\gamma_t \leftarrow \bar{\alpha}_t(1 - \bar{\alpha}_t)^2$; `// Time-dependent guidance weight`
  $\mathcal{Z}^t \leftarrow q(\mathcal{Z}^t \mid \mathcal{Z}^0, \mathbf{S}_{\mathrm{ig}}, t, \epsilon)$; `// Guided forward diffusion`
  $\hat{\epsilon} \leftarrow \epsilon_\theta(\mathcal{Z}^t, \mathcal{G}^b, t)$; `// Noise prediction`
  $\mathcal{L}_{\mathrm{LDM}} \leftarrow \|\epsilon - \hat{\epsilon}\|_2^2$; `// Calculate loss`
  $\theta \leftarrow \mathrm{optimizer}(\mathcal{L}_{\mathrm{LDM}}; \theta)$;
return ϵ_θ;

Algorithm 4: Inference Algorithm of PepGuide

Input : binding-site graph $\mathcal{G}^b$, trained $\mathcal{D}_\phi$, trained f_{IANet}, trained ϵ_θ, reference number K, candidate number N

Output : generated peptide candidates $\{\hat{\mathcal{G}}_n^p\}_{n=1}^N$

$\mathcal{C}_{\mathrm{ref}} \leftarrow \emptyset$;
for $k = 1$ **to** K **do**
  $\mathcal{Z}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  for $t = T$ **to** 1 **do**
    $\mathcal{Z}^{t-1} \leftarrow p_\theta(\mathcal{Z}^{t-1} \mid \mathcal{Z}^t, \mathcal{G}^b)$; // Unguided reverse sampling
  $\hat{\mathcal{G}}_k^p \leftarrow \mathcal{D}_\phi(\mathcal{Z}^0, \mathcal{G}^b)$; // Decode reference candidate
  $m_k \leftarrow \mathrm{PPL}(\hat{\mathcal{G}}_k^p)$; // Reference scoring
  $\mathcal{C}_{\mathrm{ref}} \leftarrow \mathcal{C}_{\mathrm{ref}} \cup \{(\hat{\mathcal{G}}_k^p, m_k)\}$;
$\mathcal{G}^{p,\mathrm{ref}} \leftarrow \arg\min_{\hat{\mathcal{G}}_k^p \in \mathcal{C}_{\mathrm{ref}}} m_k$; // Select reference peptide
$\mathbf{S}_{\mathrm{ig}} \leftarrow f_\phi(f_{\mathrm{IANet}}(\mathcal{G}^b, \mathcal{G}^{p,\mathrm{ref}}))$; // Estimate guidance signal
for $n = 1$ **to** N **do**
  $\mathcal{Z}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  for $t = T$ **to** 1 **do**
    $\gamma_t \leftarrow \bar{\alpha}_t(1 - \bar{\alpha}_t)^2$;
    $\mathcal{Z}^{t-1} \leftarrow p_\theta(\mathcal{Z}^{t-1} \mid \mathcal{Z}^t, \mathcal{G}^b, \mathbf{S}_{\mathrm{ig}}, \gamma_t)$; // Guided reverse sampling
  $\hat{\mathcal{G}}_n^p \leftarrow \mathcal{D}_\phi(\mathcal{Z}^0, \mathcal{G}^b)$; // Decode final peptide
return $\{\hat{\mathcal{G}}_n^p\}_{n=1}^N$;