PEPRAE: ADAPTING PROTEIN REPRESENTATIONS FOR TARGET-CONDITIONED PEPTIDE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing peptides for protein targets requires jointly generating amino-acid sequences and their bound three-dimensional conformations. Protein sequence and structure models provide protein-model representations, and the shared amino-acid alphabet of proteins and peptides makes these representations a natural starting point for peptide design. Yet these representations are not learned to serve as generative variables for target-conditioned peptide generation. This raises the question of how they can be adapted for this task. We introduce **PepRAE**, which performs this adaptation by learning feature latents from protein-model representations, while Cα-based spatial latents explicitly represent the peptide's bound geometry. The feature latents are learned through two reconstruction objectives. Representation reconstruction encourages the feature latents to support reconstruction of the protein-model representations. Peptide reconstruction provides downstream supervision by decoding peptide sequence and bound conformation from reconstructed features and spatial latents. After reconstruction training, target-conditioned latent diffusion jointly models peptide feature and spatial latents. Reverse diffusion followed by decoding generates peptide sequences and bound conformations. With Boltz-2 as the representation source, PepRAE achieves the highest amino-acid recovery, the lowest structural RMSDs, and the lowest Rosetta interface score (ΔG) among the evaluated baselines on the LNR benchmark. To assess dependence on the representation source, we instantiate PepRAE with the pretrained sequence model ESM-2 and with LightEnc, our lightweight sequence encoder trained from scratch. Across the three separately trained variants, peptide reconstruction remains accurate and generation performance is broadly comparable. Together, these results show that the same formulation can learn feature latents from distinct protein-model representations for target-conditioned peptide design.

1 INTRODUCTION

Peptide ligands are used as biological probes and are studied as therapeutic modalities (Muttenthaler et al., 2021). Their interactions with protein targets depend on both amino-acid identity and the arrangement of the peptide at the binding site (London et al., 2010). Following prior formulations of de novo target-conditioned peptide design, we jointly generate a peptide sequence and its bound conformation for a given protein binding site (Li et al., 2024; Kong et al., 2024; 2025).

Under this formulation, existing peptide generators model sequence and structural variables directly or through learned molecular latents (Li et al., 2024; Lin et al., 2024; Kong et al., 2024; 2025). Advances in protein sequence modeling and structure prediction have yielded protein-model representations (Lin et al., 2023; Passaro et al., 2025). The shared amino-acid alphabet of proteins and peptides makes these representations a natural starting point for peptide design. However, these representations are learned for the original objectives of protein models, rather than specifically as generative variables for target-conditioned peptide generation. This motivates our central question: how can protein-model representations be adapted into generative variables for target-conditioned peptide generation?

We introduce **PepRAE** to learn feature latents from representations supplied by frozen protein models under two complementary reconstruction objectives. Representation reconstruction encourages the feature latents to support reconstruction of the protein-model representations. Peptide reconstruction provides downstream supervision that adapts the same latents for peptide decoding.

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.

To model the peptide's bound conformation explicitly, PepRAE uses Cα-based spatial latents. During peptide reconstruction, the peptide decoder combines features reconstructed from the feature latents with the supplied target graph and spatial latents to recover the peptide sequence and bound conformation. After reconstruction training, target-conditioned latent diffusion jointly models peptide feature and spatial latents. Reverse diffusion followed by the same decoding process generates new peptide sequences and bound conformations.

With Boltz-2 as the representation source, PepRAE achieves the highest amino-acid recovery, the lowest structural RMSDs, and the lowest Rosetta interface score (ΔG) among the evaluated baselines on the LNR benchmark (Tsaban et al., 2022). To assess dependence on the representation source, we instantiate PepRAE with the pretrained sequence model ESM-2 and with LightEnc, our lightweight sequence encoder trained from scratch. Across the three separately trained variants, peptide reconstruction remains accurate and generation performance is broadly comparable.

Our contributions are as follows:

- We introduce PepRAE, a representation-adaptation framework that learns feature latents from protein-model representations through two complementary reconstruction objectives: representation reconstruction and peptide reconstruction. Target-conditioned diffusion jointly models these feature latents and Cα-based spatial latents for peptide generation.
- With Boltz-2 as the representation source, PepRAE achieves the highest amino-acid recovery, the lowest structural RMSDs, and the lowest Rosetta interface score (ΔG) among the evaluated methods on the LNR benchmark.
- We evaluate the same formulation with distinct representation sources. Across separately trained Boltz-2, ESM-2, and LightEnc variants, peptide reconstruction remains accurate and generation performance is broadly comparable.

2 Related Work

Joint peptide sequence–structure generation. Target-conditioned peptide generators jointly model amino-acid sequences and bound conformations. DiffPepBuilder uses diffusion; PepFlow models sequence and full-atom structural variables with flow matching; and PPFlow models peptide geometry through torsional flow matching (Wang et al., 2024; Li et al., 2024; Lin et al., 2024). PepGLAD learns geometric latents from peptide sequence and structure, while UniMoMo encodes molecular blocks for latent generation (Kong et al., 2024; 2025). PepTri and PepFGLD further explore task-specific guidance and flexibility-aware modeling (Nguyen et al., 2026; Niu et al., 2026). PepRAE addresses the same task but learns feature latents from representations supplied by frozen protein models.

Protein-model representations in biomolecular generation. DiMA and PepEDiff use protein-model representation spaces for protein and peptide sequence generation, respectively (Meshchaninov et al., 2025; Liang et al., 2026). CHEAP compresses ESMFold representations, and PLAID models protein sequence–structure generation in the resulting latent space (Lu et al., 2025; 2026). PRIME is a recent approach that uses a frozen pretrained EPT encoder followed by learned projections to molecular latents for peptide and antibody CDR generation (Tian et al., 2026; Jiao et al., 2026b). PepRAE learns feature latents from protein-model representations through representation reconstruction and peptide reconstruction.

Representation autoencoding and adaptation. Image-generation methods provide related formulations for learning generative latents from pretrained representations. RAE uses pretrained visual representations as generative latent spaces, while FAE learns compact latents that reconstruct pretrained features before image decoding (Zheng et al., 2026; Gao et al., 2026). RepTok studies representation adaptation and decoding with different pretrained visual encoders (Gui et al., 2026). These works address representation adaptation for generation, whereas target-conditioned peptide design requires generating peptide sequences together with their bound conformations. In PepRAE, peptide reconstruction adapts the feature latents by decoding peptides from reconstructed source features, Cα-based spatial latents, and the supplied target graph. Target-conditioned diffusion then jointly models the feature and spatial latents for generation.

3 METHOD

3.1 TASK AND FRAMEWORK OVERVIEW

Given a protein target, its binding site, and a specified peptide length, PepRAE jointly generates a peptide sequence and its bound conformation. We first adapt representations from frozen protein models into feature latents through representation reconstruction and peptide reconstruction. Features reconstructed from the feature latents are used for peptide decoding together with the supplied target graph and Cα-based spatial latents (Sections 3.2 and 3.3). After reconstruction training, target-conditioned diffusion jointly models peptide feature and spatial latents. Reverse diffusion followed by the same decoders generates peptide sequences and bound conformations (Section 3.4). Figure 1 summarizes the framework.

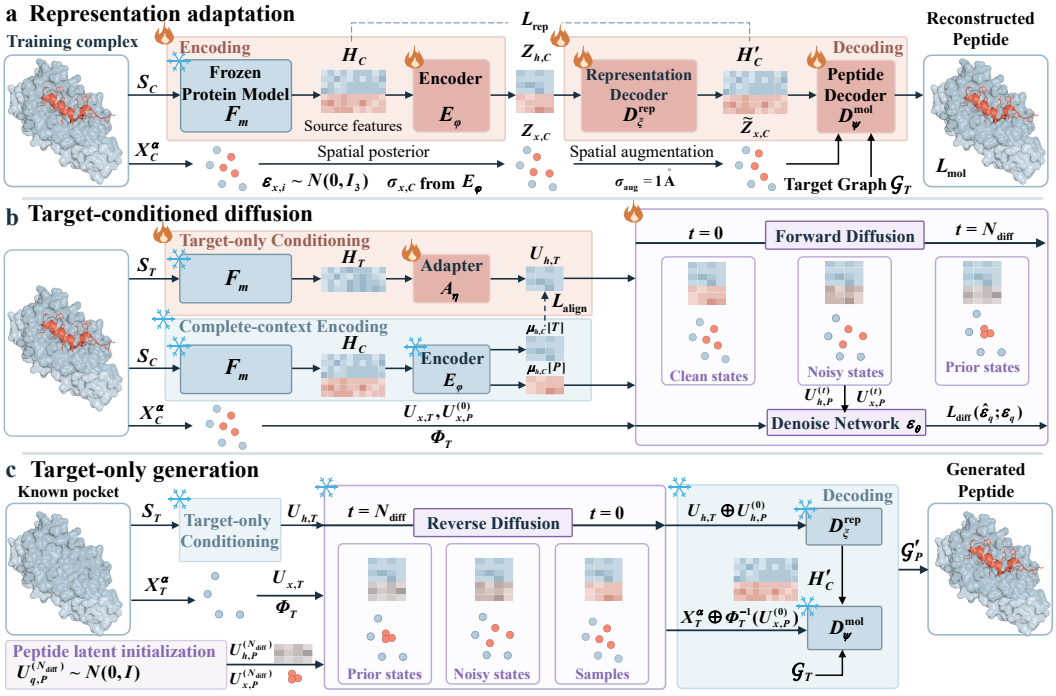


Figure 1: **Overview of PepRAE.** Here $\boldsymbol{Z}$ denotes autoencoder posterior samples, whereas $\boldsymbol{U}$ denotes diffusion-side quantities defined in Section 3.4. (a) Joint reconstruction learns feature latents, using Cα-anchored spatial inputs for peptide decoding. (b) Frozen-encoder peptide means provide clean feature targets; target means supervise the target-only adapter. The denoiser predicts noise in both peptide fields under fixed target conditions. (c) Reverse diffusion followed by decoding generates peptide sequences and bound conformations. Snowflakes/flames mark frozen/updated modules. Molecular views and residue counts are schematic.

3.2 PROTEIN-MODEL REPRESENTATIONS AND SOURCE FEATURES

Let T denote the target residues retained around the binding site, P the peptide residues, and $C = T \dot{\cup} P$ the complex residue set. The supplied target graph $\mathcal{G}_T$ contains residue identities, atoms, bonds, and coordinates for the retained target residues; for a training complex, $\mathcal{G}_P$ denotes the corresponding observed peptide graph.

The sequence context $\boldsymbol{S}_T$ contains the full target chains, including residues not retained in T; $\boldsymbol{S}_C$ additionally contains the peptide sequence of the training complex. Inputs to the frozen protein model are constructed from these sequence contexts and chain segmentation, while observed geometry enters separately through graph coordinates and spatial latents. We call the frozen protein model m the *source model*. A fixed source interface $\mathcal{F}_m$ prepares each context, selects and preprocesses the model's internal representations, and aligns them to the retained graph rows. Both the source model

and this interface remain fixed throughout PepRAE training. We call the resulting residue-aligned tensors *source features*:

$$\begin{aligned} \boldsymbol{H}_T^{(m)} &= \mathcal{F}_m(\boldsymbol{S}_T; T) \in \mathbb{R}^{|T| \times d_m}, \\ \boldsymbol{H}_C^{(m)} &= \mathcal{F}_m(\boldsymbol{S}_C; C) \in \mathbb{R}^{|C| \times d_m}. \end{aligned} \tag{1}$$

Here d_m is the source-feature width after preprocessing; the source superscript is omitted below. We refer to $\boldsymbol{H}_T$ as the target-only source features and to $\boldsymbol{H}_C$ as the complete-context source features. The target rows $\boldsymbol{H}_T$ and $\boldsymbol{H}_C[T]$ correspond to the same retained residues but are extracted from the respective sequence contexts.

The fixed interface defines the source features; their adaptation for peptide decoding is learned through the reconstruction objectives below. For each source model, we train a separate autoencoder and diffusion model within the same generative formulation. Appendix C specifies the source interfaces, and Appendix A.1 summarizes notation and information flow.

3.3 Representation adaptation by joint reconstruction

We use a variational autoencoder with feature latents and Cα-based spatial latents. Let $\boldsymbol{X}_C^{\alpha} = \{\boldsymbol{x}_i^{\alpha}\}_{i \in C}$ denote the observed Cα coordinates of the training complex. Each residue $i \in C$ is associated with a feature latent $\boldsymbol{z}_{h,i} \in \mathbb{R}^{d_h}$ and a spatial latent $\boldsymbol{z}_{x,i} \in \mathbb{R}^3$, whose posterior is centered at $\boldsymbol{x}_i^{\alpha}$. Representation reconstruction and peptide reconstruction jointly adapt the feature latents for peptide decoding.

Feature and spatial posteriors. The encoder operates on complete-context source features, while observed Cα coordinates specify the means of the spatial latents. The encoder normalizes $\boldsymbol{H}_C$, adds learned chain embeddings, and applies self-attention over the complex:

$$(\boldsymbol{\mu}_{h,C}, \boldsymbol{\sigma}_{h,C}, \boldsymbol{\sigma}_{x,C}) = \mathcal{E}_{\phi}(\boldsymbol{H}_C; \boldsymbol{k}_C), \tag{2}$$

where $\boldsymbol{k}_C$ records chain identities. For each residue, these parameters define a diagonal posterior over feature latents and an isotropic posterior over spatial latents:

$$\begin{aligned} q_{\phi}^{h}(\boldsymbol{z}_{h,i} \mid \boldsymbol{H}_C, \boldsymbol{k}_C) &= \mathcal{N}\big(\boldsymbol{\mu}_{h,i}, \operatorname{diag}(\boldsymbol{\sigma}_{h,i}^2)\big), \\ q_{\phi}^{x}(\boldsymbol{z}_{x,i} \mid \boldsymbol{H}_C, \boldsymbol{k}_C, \boldsymbol{x}_i^{\alpha}) &= \mathcal{N}\big(\boldsymbol{x}_i^{\alpha}, \sigma_{x,i}^2 \boldsymbol{I}_3\big). \end{aligned} \tag{3}$$

The encoder learns feature posterior means and standard deviations and spatial posterior standard deviations; observed Cα coordinates fix the spatial means. Reparameterized samples are stacked into feature and spatial matrices $\boldsymbol{Z}_{h,C}$ and $\boldsymbol{Z}_{x,C}$, respectively.

Joint source-feature and peptide reconstruction. The representation decoder maps feature latents to reconstructed source features $\boldsymbol{H}'_C$, which the peptide decoder combines with spatial latents and the supplied target graph to recover the peptide sequence and bound conformation. During training, peptide spatial rows receive an additional perturbation, giving $\widetilde{\boldsymbol{Z}}_{x,C}$, and reconstruction follows:

$$\begin{aligned} \boldsymbol{H}'_C &= \mathcal{D}_{\xi}^{\mathrm{rep}}(\boldsymbol{Z}_{h,C}), \\ \mathcal{G}'_P &= \mathcal{D}_{\psi}^{\mathrm{mol}}(\boldsymbol{H}'_C, \widetilde{\boldsymbol{Z}}_{x,C}; \mathcal{G}_T). \end{aligned} \tag{4}$$

Primes denote decoder outputs. Both reconstruction objectives train the encoder $\mathcal{E}_{\phi}$ and representation decoder $\mathcal{D}_{\xi}^{\mathrm{rep}}$; peptide reconstruction also trains the peptide decoder $\mathcal{D}_{\psi}^{\mathrm{mol}}$. Thus, peptide reconstruction adapts the feature latents for peptide decoding through $\boldsymbol{H}'_C$, while source-feature reconstruction constrains the same latents to reconstruct the fixed source features.

Peptide decoding uses a residue-block formulation with iterative equivariant coordinate updates (Kong et al., 2025) to produce amino-acid identities, atomic coordinates, and inter-residue bonds. Conditional flow matching trains coordinate reconstruction (Lipman et al., 2023), using observed atom types and supplied topology during training. At generation time, predicted residue identities define the peptide atom templates for coordinate decoding. Appendix A.2.3 details the decoding procedure and training-time spatial perturbation.

Joint reconstruction objective. We implement representation reconstruction as source-feature reconstruction against a row-normalized reference:

$$\overline{\boldsymbol{H}}_C = \mathrm{LN}_0(\boldsymbol{H}_C), \qquad \mathcal{L}_{\mathrm{rep}} = d_{\mathrm{rep}}(\boldsymbol{H}'_C, \overline{\boldsymbol{H}}_C). \tag{5}$$

Here LN_0 has no learned affine parameters, whereas the representation decoder's output LayerNorm is affine. The loss d_{rep} combines Smooth-L1 and row-wise cosine terms. Peptide reconstruction combines identity, atom-flow, bond, and local-geometry losses in $\mathcal{L}_{\mathrm{mol}}$. Adding KL regularization against $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_h})$ for the feature posterior and $\mathcal{N}(\boldsymbol{x}_i^\alpha, \boldsymbol{I}_3)$ for the spatial posterior gives

$$\mathcal{L}_{\mathrm{AE}}(r) = \mathcal{L}_{\mathrm{rep}} + \mathcal{L}_{\mathrm{mol}} + \beta_h(r)\mathcal{L}_{\mathrm{KL}}^h + \beta_x(r)\mathcal{L}_{\mathrm{KL}}^x, \tag{6}$$

where r indexes optimizer updates and the β weights schedule KL regularization. Loss weights, reductions, and auxiliary target-residue reconstruction are specified in Appendix A.2.

3.4 Target-conditioned latent diffusion

With the autoencoder frozen, we learn a target-conditioned joint distribution over peptide feature and spatial states. We use $\boldsymbol{U}$ for diffusion-side quantities, distinct from the autoencoder posterior samples $\boldsymbol{Z}$. Only peptide states evolve with diffusion time; their row count is the specified peptide length, while target conditions remain fixed.

Target-only conditioning. Complete-context target means depend on the peptide and are unavailable before generation. An adapter instead maps target-only source features to $\boldsymbol{U}_{h,T} = \mathcal{A}_\eta(\boldsymbol{H}_T)$, supervised by the frozen encoder's complete-context target means:

$$\mathcal{L}_{\mathrm{align}} = d_{\mathrm{align}}(\boldsymbol{U}_{h,T}, \boldsymbol{\mu}_{h,C}[T]), \tag{7}$$

where d_{align} combines mean-squared error and cosine terms. These means serve only as alignment targets; denoising and generation use the adapter output.

Peptide states and denoising. Spatial diffusion states use a target-centered, scaled coordinate frame:

$$\Phi_T(\boldsymbol{X}) = \frac{\boldsymbol{X} - \mathbf{1}\boldsymbol{c}_T^\top}{\kappa_x}, \qquad \Phi_T^{-1}(\boldsymbol{Y}) = \kappa_x \boldsymbol{Y} + \mathbf{1}\boldsymbol{c}_T^\top, \tag{8}$$

where $\boldsymbol{c}_T$ is the mean Cα coordinate of the retained target residues and $\kappa_x = 10$ Å (Luo et al., 2022; Kong et al., 2025). The fixed target condition is $\mathcal{U}_T = (\boldsymbol{U}_{h,T}, \boldsymbol{U}_{x,T})$, with $\boldsymbol{U}_{x,T} = \Phi_T(\boldsymbol{X}_T^\alpha)$. For diffusion training, both clean peptide states are constructed from posterior means. The feature mean is used directly in the autoencoder feature-latent coordinate system, while the spatial mean $\boldsymbol{X}_P^\alpha$ is transformed by Φ_T:

$$\boldsymbol{U}_{h,P}^{(0)} = \boldsymbol{\mu}_{h,C}[P], \qquad \boldsymbol{U}_{x,P}^{(0)} = \Phi_T(\boldsymbol{X}_P^\alpha). \tag{9}$$

For each field $q \in \{h, x\}$, forward diffusion adds independent Gaussian noise:

$$\boldsymbol{U}_{q,P}^{(t)} = \alpha_t \boldsymbol{U}_{q,P}^{(0)} + \sigma_t \boldsymbol{\epsilon}_{q,t}, \qquad \boldsymbol{\epsilon}_{q,t} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{10}$$

Here t is diffusion time, and α_t, σ_t follow the noise schedule. The equivariant denoiser ϵ_θ takes the joint peptide state $\mathcal{U}_P^{(t)} = (\boldsymbol{U}_{h,P}^{(t)}, \boldsymbol{U}_{x,P}^{(t)})$ and predicts $\widehat{\boldsymbol{\epsilon}}_{h,t}$ and $\widehat{\boldsymbol{\epsilon}}_{x,t}$ under fixed target-side conditioning.

Diffusion objective. The denoising loss $\mathcal{L}_{\mathrm{diff}}$ is a weighted mean-squared error between predicted and sampled noise. Together with alignment, it gives

$$\mathcal{L}_{\mathrm{LDM}} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}}. \tag{11}$$

Denoising uses a stop-gradient copy of $\boldsymbol{U}_{h,T}$, so $\mathcal{L}_{\mathrm{diff}}$ updates the denoiser while $\mathcal{L}_{\mathrm{align}}$ trains the adapter. Appendix A.3 specifies loss weights and diffusion schedules.

Sampling and decoding. Generation starts at step N_{diff} with independent Gaussian noise, $\boldsymbol{U}_{q,P}^{(N_{\text{diff}})} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $q \in \{h, x\}$, and applies reverse diffusion under fixed target conditions. The resulting $\boldsymbol{U}_{q,P}^{(0)}$ are decoded using the trained decoders. $\boldsymbol{U}_{h,T}$ and $\boldsymbol{U}_{h,P}^{(0)}$ enter the representation decoder directly in its feature-latent coordinates, while Φ_T^{-1} restores the peptide spatial state to physical coordinates:

$$\begin{aligned} \boldsymbol{H}'_C &= \mathcal{D}_\xi^{\text{rep}}(\boldsymbol{U}_{h,T} \oplus \boldsymbol{U}_{h,P}^{(0)}), \\ \mathcal{G}'_P &= \mathcal{D}_\psi^{\text{mol}}\big(\boldsymbol{H}'_C, \boldsymbol{X}_T^{\alpha} \oplus \Phi_T^{-1}(\boldsymbol{U}_{x,P}^{(0)}); \mathcal{G}_T\big). \end{aligned} \tag{12}$$

Here $\oplus$ assembles target and peptide rows in complex order. The supplied target remains fixed as the decoders produce the peptide sequence and bound conformation. Appendix A.4 gives the training and generation algorithms.

4 Experiments

4.1 Experimental setup

Data and split. We train on PepDataBench, constructed from PepMerge (Li et al., 2024) and the supervised target–peptide subset of PepBench (Kong et al., 2024). Peptides are restricted to 3–25 residues; applying the same criterion to LNR (Tsaban et al., 2022) leaves 92 evaluation targets. Following the clustering strategy of PepFlow (Li et al., 2024), target sequences from the curated pool and the filtered LNR set are jointly clustered with MMseqs2 (Steinegger & Söding, 2017) at 40% sequence identity and 80% coverage. Complexes in clusters containing an LNR target are excluded, and the remaining target clusters are partitioned into training and validation, yielding 9,113 training and 1,064 validation complexes. Appendix B details dataset construction and splitting.

Models and training. We use Boltz-2, ESM-2, and LightEnc as frozen representation sources (Lin et al., 2023; Passaro et al., 2025). We use a feature-latent width of $d_h = 8$ throughout. For the three-source comparison, source-specific preprocessing yields features of width $d_m = 384$, while the downstream PepRAE architecture, objectives, and optimization settings are held fixed. For the primary benchmark, source comparison, and main ablations, one validation-selected autoencoder is held fixed within each configuration while three diffusion models are trained from independent initializations. Generation metrics are computed separately for each diffusion run and then averaged across the three runs. Source extraction and training settings are given in Appendices C and J.2.

Comparisons. The primary benchmark compares the Boltz-2 PepRAE configuration with PepGLAD, PepFlow, PepTri, UniMoMo, and PRIME (Kong et al., 2024; Li et al., 2024; Nguyen et al., 2026; Kong et al., 2025; Tian et al., 2026). Each baseline is retrained once on the common PepDataBench training split using its released implementation and method-specific settings. Appendix E provides baseline training and sampling details and additional benchmark results.

Generation evaluation. Evaluation uses the filtered LNR benchmark with reference-defined binding sites, bound target coordinates, and reference peptide lengths. Each model generates 100 candidates per target. Following UniMoMo and PepTri (Kong et al., 2025; Nguyen et al., 2026), all methods use the same metric-specific candidate-pool aggregation. Within each target, AAR, $F1_{\text{intra}}$, and DockQ use maxima, whereas C-RMSD, L-RMSD, M-RMSD, and Rosetta ΔG use minima. The per-target extrema are averaged across targets, except that RMSDs use the median of per-target minima. Metrics are reduced independently, so different reported columns need not correspond to the same candidate; they characterize recovery over the generated candidate pool rather than a single selected design.

AAR measures sequence recovery; C-RMSD measures target-relative Cα recovery, while L-RMSD and M-RMSD measure peptide-aligned Cα and mapped-atom recovery. $F1_{\text{intra}}$ and DockQ measure contact and interface recovery, and Rosetta ΔG provides an empirical interface score. IMP is the percentage of targets for which at least one generated candidate has a Rosetta ΔG no higher than that of the corresponding native complex. D_{seq} and D_{pose} measure diversity. Appendix D defines all metrics and aggregation rules.

4.2 TARGET-CONDITIONED PEPTIDE DESIGN ON LNR

Table 1 compares the Boltz-2 PepRAE configuration with the evaluated baselines under the common protocol. PepRAE achieves the highest reported AAR, the lowest C-RMSD, L-RMSD, and M-RMSD, and the lowest Rosetta ΔG and the highest IMP, while ranking second to UniMoMo in $F1_{\text{intra}}$ and DockQ. Sequence diversity is comparable to PepGLAD and PepTri, while pose diversity is lower than that of most baselines under the benchmark clustering metric.

Table 1: **Target-conditioned peptide design on LNR.** All methods use the same 100-candidate-per-target, metric-specific aggregation protocol, applied independently across metrics. PepRAE reports means of three diffusion-run statistics sharing one selected Boltz-2 autoencoder; baselines are single retrained models. Boldface and underlining mark the best and second-best values. ΔG is in REU.

Method	Recovery				Contact / Interface		Rosetta		Diversity	
	AAR↑ (%)	C-RMSD (Å)↓	L-RMSD (Å)↓	M-RMSD (Å)↓	$F1_{\text{intra}}$ ↑	DockQ↑	ΔG ↓	IMP↑ (%)	D_{seq} ↑	D_{pose} ↑
PepGLAD	37.47	3.03	1.83	1.84	0.703	0.598	-22.03	16.30	<u>0.655</u>	0.673
PepFlow	32.89	3.18	1.74	1.80	0.767	0.566	-21.05	14.13	0.588	0.489
PepTri	<u>39.23</u>	3.15	1.73	1.80	0.868	0.657	-15.69	10.87	**0.663**	**0.782**
UniMoMo	37.59	<u>2.61</u>	<u>1.30</u>	<u>1.51</u>	**0.906**	**0.749**	<u>-28.70</u>	<u>28.26</u>	0.628	0.651
PRIME	38.06	2.78	1.42	1.75	0.880	0.727	-27.83	26.09	0.455	<u>0.763</u>
PepRAE	**39.73**	**2.43**	**1.21**	**1.47**	<u>0.895</u>	<u>0.729</u>	**-30.16**	**29.35**	0.642	0.560

PepRAE also achieves the highest TM-score, lowest clash rates, and lowest side-chain dihedral-distribution distance, while its co-design diversity is lower than that of most baselines (Table 10). Appendix E reports length-stratified results, amino-acid composition, run variation, and the supplementary single-run comparison.

4.3 RECONSTRUCTION AND LATENT-SPACE ANALYSIS

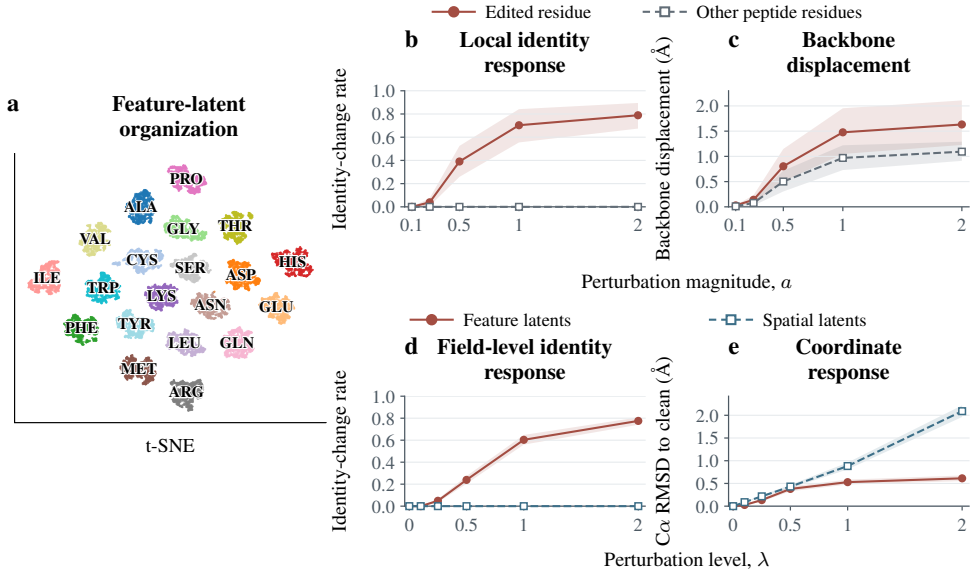


Figure 2: **Feature-latent organization and perturbation responses.** (a) t-SNE of validation peptide feature posterior means, colored by amino-acid identity. (b,c) Single-residue feature-latent perturbations. (d,e) Peptide-wide feature- or spatial-field perturbations. Perturbation magnitudes are not norm-matched across fields. Curves show means across targets; shading denotes pointwise 95% PDB-accession-cluster bootstrap intervals.

Using deterministic posterior means on the 1,064 validation complexes, the selected Boltz-2 autoencoder achieves peptide-row source-feature cosine similarity of 0.779, residue accuracy of 99.89%, and peptide-aligned Cα RMSD of 0.474 Å. Reconstruction uses the supplied target graph and observed Cα positions as spatial posterior means. Cosine similarity measures source-feature fidelity; the latter two metrics assess peptide reconstruction. Appendix H reports full metrics across autoencoder seeds.

Figure 2a visualizes identity-associated groups in t-SNE. Cross-complex kNN@20 purity of 0.9108 and held-out-target linear-probe accuracy of 99.66%, both measured in the original feature-latent space, support local identity organization and linear readability.

We compare perturbed decoder outputs with paired unperturbed decoding from deterministic peptide posterior means, holding other inputs and decoder randomness fixed. Coordinate responses use the fixed target frame without peptide superposition. Single-residue feature perturbations produce localized top-1 identity changes, with no observed top-1 changes elsewhere, while backbone displacements extend to other peptide residues (Figure 2b,c).

Peptide-wide feature perturbations affect both identities and coordinates; spatial perturbations change coordinates without observed top-1 identity changes (Figure 2d,e). This pattern supports a more geometry-specific spatial response over the tested range; because magnitudes are not norm-matched, we compare response types rather than absolute sensitivity. Together with the readouts, these interventions indicate that feature latents contain accessible residue identity and influence both identity and coordinate decoding. Protocols are given in Appendix I.

4.4 Ablation studies

We test representation-adaptation and target-adapter training. Tables report three-run generation means; Appendix G gives intervention details and run-to-run variation.

4.4.1 Representation-adaptation training

We separately test representation reconstruction and peptide supervision of the representation decoder. *No* $\mathcal{L}_{\text{rep}}$ removes representation reconstruction, retaining peptide reconstruction, KL regularization, and the forward pathway. The $\mathcal{L}_{\text{rep}}$-only $\mathcal{D}_\xi^{\text{rep}}$ control retains both reconstruction objectives but prevents peptide loss from updating representation-decoder parameters; peptide-loss gradients still propagate through this decoder to the feature latents and encoder.

Without $\mathcal{L}_{\text{rep}}$, residue accuracy remains nearly unchanged, but peptide-row source-feature cosine similarity falls from 0.779 to approximately zero and peptide-aligned RMSDs increase (Appendix G.1). Thus, accurate identity reconstruction does not ensure source-feature fidelity. Both controls increase mean RMSDs and Rosetta ΔG and decrease mean AAR, DockQ, and IMP (Table 2), supporting representation reconstruction and supervision of the representation decoder by both reconstruction objectives.

Table 2: **Generation under representation-adaptation ablations.** Three-run means; reconstruction is in Table 13. Boldface/underlining mark the best/second-best means. RMSDs are in Å, ΔG in REU, and AAR/IMP in percent.

Configuration	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	ΔG ↓	IMP↑
PepRAE	**39.73%**	**2.432**	**1.213**	**1.474**	**0.729**	**-30.161**	**29.35%**
No $\mathcal{L}_{\text{rep}}$	<u>38.91%</u>	<u>2.612</u>	1.343	<u>1.493</u>	<u>0.724</u>	-27.946	25.00%
$\mathcal{L}_{\text{rep}}$-only $\mathcal{D}_\xi^{\text{rep}}$	38.63%	2.682	<u>1.330</u>	1.523	0.717	<u>-28.692</u>	<u>26.81%</u>

4.4.2 Target-adapter training

With the autoencoder fixed, PepRAE aligns the adapter output from target-only source features to complete-context target feature posterior means and stops denoising gradients at the adapter output. *No supervision* removes alignment while retaining this gradient stop; *Denoising-supervised* removes alignment and allows denoising gradients to update the adapter. Both controls retain the fixed target

spatial condition and supplied target graph. All seven generation means favor explicit alignment over either alternative under the evaluated training setup (Table 3).

Table 3: **Target-adapter training ablations.** Three-run means with one fixed autoencoder; best and second-best means are bold and underlined. RMSDs are in Å, ΔG in REU, and AAR/IMP in percent.

Configuration	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	$\Delta G\downarrow$	IMP↑
PepRAE	**39.73%**	**2.432**	**1.213**	**1.474**	**0.729**	**-30.161**	**29.35%**
No supervision	37.74%	<u>2.767</u>	<u>1.343</u>	1.586	<u>0.716</u>	-26.215	23.55%
Denoising-supervised	<u>38.02%</u>	2.780	1.382	<u>1.547</u>	0.701	<u>-27.583</u>	<u>25.72%</u>

4.5 Representation adaptation across sources

We train separate PepRAE variants using Boltz-2, ESM-2, and LightEnc, with matched latent widths, reconstruction objectives, downstream architecture, and optimization settings. All three support accurate peptide reconstruction, with residue accuracy of 99.89–100.00% and peptide-aligned Cα RMSDs of 0.471–0.492 Å (Table 17).

Table 4: **Generation across representation sources.** Three-run means per source-specific autoencoder; Boltz-2 reuses the benchmark runs. RMSDs are in Å, ΔG in REU, and AAR/IMP in percent.

Source	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	$\Delta G\downarrow$	IMP↑
Boltz-2	39.73%	2.432	1.213	1.474	0.729	−30.161	29.35%
ESM-2	40.41%	2.528	1.283	1.499	0.727	−30.136	28.26%
LightEnc	38.15%	2.485	1.187	1.408	0.746	−30.269	28.62%

Generation metrics favor different sources (Table 4): ESM-2 has the highest mean AAR, Boltz-2 the lowest C-RMSD, and LightEnc the highest DockQ, with no source leading across all metrics. Thus, the formulation supports accurate reconstruction and target-conditioned generation with pretrained sources and with LightEnc, trained from scratch. Peptide reconstruction and generation are compared across sources using the same metrics within each task; source-feature reconstruction is evaluated against each source's own features. Appendix F reports run variation.

5 Conclusion and Scope

PepRAE adapts protein-model representations into feature latents for target-conditioned peptide sequence–structure generation through representation and peptide reconstruction, while Cα-based spatial latents represent bound geometry. A target adapter supplies the feature component of target-only conditioning for joint diffusion of peptide feature and spatial latents.

With Boltz-2, PepRAE achieves the highest amino-acid recovery, lowest structural RMSDs, and lowest Rosetta interface score (ΔG) among evaluated baselines on LNR. Separately trained Boltz-2, ESM-2, and LightEnc variants support accurate reconstruction and target-conditioned generation.

Our evaluation covers canonical peptides of prescribed length at known binding sites with bound target coordinates. Native-independent candidate selection, unbound conformations, broader peptide classes, and experimental binding measurements remain complementary extensions.

AI USE STATEMENT

Generative AI was used to assist with language refinement, manuscript organization and literature search. AI tools were not used to generate research datasets or experimental measurements. The authors reviewed all AI-assisted content and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Sections 3–4 describe the model, training, and evaluation setup. Appendices A, B, C, and D provide additional method details, dataset construction, source-feature extraction, and metric definitions. Appendix J records architecture, optimization, checkpoint-selection, and run-seed settings, while Appendix F reports the matched source-comparison protocol and run-level results.

ETHICS STATEMENT

This study is entirely computational and uses existing structural data. It does not involve human participants or newly conducted animal or laboratory experiments. We are not aware of study-specific ethical concerns.

REFERENCES

Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, Jason W. Labonte, Michael S. Pacella, Richard Bonneau, Philip Bradley, Roland L. Dunbrack, Jr, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017. doi:10.1021/acs.jctc.7b00125.

Sankar Basu and Björn Wallner. DockQ: A quality measure for protein–protein docking models. *PLOS ONE*, 11(8):e0161879, 2016. doi:10.1371/journal.pone.0161879.

Suhaas Bhat, Kalyan Palepu, Lauren Hong, Joey Mao, Tianzheng Ye, Rema Iyer, Lin Zhao, Tianlai Chen, Sophia Vincoff, Rio Watson, Tian Z. Wang, Divya Srijay, Venkata Srikar Kavirayuni, Kseniia Kholina, Shrey Goel, Pranay Vure, Aniruddha J. Deshpande, Scott H. Soderling, Matthew P. DeLisa, and Pranam Chatterjee. De novo design of peptide binders to conformationally diverse targets with contrastive language modeling. *Science Advances*, 11(4):eadr8638, 2025. doi:10.1126/sciadv.adr8638.

Leo Tianlai Chen, Zachary Quinn, Madeleine Dumas, Christina Peng, Lauren Hong, Moises Lopez-Gonzalez, Alexander Mestre, Rio Watson, Sophia Vincoff, Lin Zhao, Jianli Wu, Audrey Stavrand, Mayumi Schaepers-Cheu, Tian Zi Wang, Divya Srijay, Connor Monticello, Pranay Vure, Rishab Pulugurta, Sarah Pertsemlidis, Kseniia Kholina, Shrey Goel, Matthew P. DeLisa, Jen-Tsan Ashley Chi, Ray Truant, Hector C. Aguilar, and Pranam Chatterjee. Target sequence-conditioned design of peptide binders using masked language modeling. *Nature Biotechnology*, 44(6):1002–1010, 2026. doi:10.1038/s41587-025-02761-2.

Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009. doi:10.1093/bioinformatics/btp163.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Robbert J. de Haas, Neville Bethel, Philip J. Y. Leung, Timothy F. Huddy, Sam Pellock, Doug Tischer, Frederick Chan, Brian Koepnick, Hannah Nguyen, Alex Kang, Banumathi Sankaran, Asim K. Bera, Neil P. King, and David Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi:10.1126/science.add2187.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/639d992f819c2b40387d4d5170b8ffd7-Abstract-Conference.html.

Wenze Ding. Computationally evidence-grounded sequence-first design of peptide binders. *Advanced Science*, pp. e77125, 2026. doi:10.1002/advs.77125.

Yuan Gao, Chen Chen, Tianrong Chen, and Jiatao Gu. One layer is enough: Adapting pretrained visual encoders for image generation. In *CVPR*, 2026. URL https://arxiv.org/abs/2512.07829.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-Proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6e1734c47c0cc899021060d88f69dc65-Abstract-Conference.html.

Ming Gui, Johannes Schusterbauer, Timy Phan, Felix Krause, Joshua Susskind, Miguel Angel Bautista, and Björn Ommer. Adapting self-supervised representations as a latent space for efficient generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 124272–124291, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c9c988de81687d84e05616c0182c4308-Paper-Conference.pdf.

Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. doi:10.1073/pnas.89.22.10915.

Rui Jiao, Xiangzhe Kong, Yinjun Jia, Yijia Zhang, Ziyi Yang, Yang Liu, and Jianzhu Ma. Scalable peptide design via memory-efficient equivariant transformer. *arXiv preprint arXiv:2606.25006*, 2026a. URL https://arxiv.org/abs/2606.25006.

Rui Jiao, Xiangzhe Kong, Li Zhang, Ziyang Yu, Fangyuan Ren, Wenjuan Tan, Wenbing Huang, and Yang Liu. An equivariant pretrained transformer for unified 3D molecular representation learning. *Nature Communications*, 17:2606, 2026b. doi:10.1038/s41467-026-69185-7. URL https://www.nature.com/articles/s41467-026-69185-7.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-atom peptide design with geometric latent diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74808–74839, 2024. doi:10.52202/079017-2379.

Xiangzhe Kong, Zishen Zhang, Ziting Zhang, Rui Jiao, Jianzhu Ma, Wenbing Huang, Kai Liu, and Yang Liu. UniMoMo: Unified generative modeling of 3D molecules for de novo binder design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31397–31418, 2025. URL https://proceedings.mlr.press/v267/kong25b.html.

Xiangzhe Kong, Junwei Chen, Ziting Zhang, Gaodeng Li, Qingyuan Zhu, Lei Wei, Mingyu Li, Yan Shi, Weiyang Dai, Zishen Zhang, Wenjuan Tan, Rui Jiao, Xiaolun Wang, Jiqing Zheng, Ziyang Yu, Qilong Wu, Zhiye Guo, Li Zhang, Wentao Li, Qiaojing Huang, Tian Zhu, Xiaowo Wang, Wenbing Huang, Yuli She, Jian Zhang, Yang Liu, Kai Liu, and Jianzhu Ma. Programming biomolecular interactions with all-atom generative model. *bioRxiv*, 2026. doi:10.64898/2026.03.12.711044.

Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-atom peptide design based on multi-modal flow matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27615–27640, 2024. URL https://proceedings.mlr.press/v235/li24o.html.

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. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1a0672689a693e0764f93f900488b3d9-Abstract-Conference.html.

Qiuzhen Li, Efstathios Nikolaos Vlachos, and Patrick Bryant. Design of linear and cyclic peptide binders from protein sequence information. *Communications Chemistry*, 8:211, 2025b. doi:10.1038/s42004-025-01601-3.

Po-Yu Liang, Tibo Duran, and Jun Bai. PepEDiff: Out-of-distribution sampling peptide binder design via protein embedding diffusion. In *Proceedings of the 17th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–10, 2026. doi:10.1145/3807503.3819492.

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

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi:10.1126/science.ade2574.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Nir London, Dana Movshovitz-Attias, and Ora Schueler-Furman. The structural basis of peptide–protein binding strategies. *Structure*, 18(2):188–199, 2010. doi:10.1016/j.str.2009.11.012.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Amy X. Lu, Wilson Yan, Kevin K. Yang, Vladimir Gligorijevic, Kyunghyun Cho, Pieter Abbeel, Richard Bonneau, and Nathan C. Frey. Tokenized and continuous embedding compressions of protein sequence and structure. *Patterns*, 6(6):101289, 2025. doi:10.1016/j.patter.2025.101289.

Amy X. Lu, Jason Yang, Yueming Long, Theophile Lambert, Wilson Yan, Sarah A. Robinson, Kevin K. Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Nathan C. Frey, Frances Arnold, and Pieter Abbeel. Controllable all-atom protein generation with latent diffusion. *bioRxiv*, 2026. doi:10.1101/2024.12.02.626353. URL https://www.biorxiv.org/content/10.1101/2024.12.02.626353v3. Version 3, posted January 7, 2026.

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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9754–9767, 2022. doi:10.52202/068431-0709. URL https://papers.nips.cc/paper_files/paper/2022/hash/3fa7d76a0dc1179f1e98d1bc62403756-Abstract-Conference.html.

Qingyi Mao, Silong Zhai, Sen Cao, Renjie Zhu, Wen Xu, Chengyun Zhang, Ning Zhu, Jingjing Guo, and Hongliang Duan. High-PepBinder: A pLM-guided latent diffusion framework for affinity-aware target-specific peptide design. *Journal of Chemical Information and Modeling*, 66 (8):4863–4877, 2026. doi:10.1021/acs.jcim.6c00253.

Viacheslav Meshchaninov, Pavel Strashnov, Andrey Shevtsov, Fedor Nikolaev, Nikita Ivanisenko, Olga Kardymon, and Dmitry Vetrov. Diffusion on language model encodings for protein sequence

generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43837–43878, 2025. URL https://proceedings.mlr.press/v267/meshchaninov25a.html.

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.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3): 443–453, 1970. doi:10.1016/0022-2836(70)90057-4.

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 *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2dbb8bfe4cd3875609b23799830ee865-Abstract-Conference.html.

Dongjiang Niu, Xiaofeng Wang, Zhiqiang Wei, and Zhen Li. Flexibility-aware geometric latent diffusion for full-atom peptide design. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kH2KFnlxmU.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, Yehlin Cho, Kourosh H. Ghamary, Laura Vinué, Brahm J. Yachnin, Andrew M. Wollacott, Stephen Buckley, Adrie H. Westphal, Simon Lindhoud, Sandrine Georgeon, Casper A. Goverde, Georgios N. Hatzopoulos, Pierre Gönczy, Yannick D. Muller, Gerald Schwank, Daan C. Swarts, Alex J. Vecchio, Bernard L. Schneider, Sergey Ovchinnikov, and Bruno E. Correia. One-shot design of functional protein binders with BindCraft. *Nature*, 646(8084):483–492, 2025. doi:10.1038/s41586-025-09429-6.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi:10.1101/2025.06.14.659707.

Hao Qian, Shikui Tu, and Lei Xu. Full-atom peptide design via Riemannian–Euclidean Bayesian Flow Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(29):24918–24926, 2026. doi:10.1609/aaai.v40i29.39679.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, Mateo Reveiz, Roman Bushuiev, Tally Portnoi, Tomáš Pluskal, Josef Sivic, Karsten Kreis, Arash Vahdat, Shamayeeta Ray, Jonathan T. Goldstein, Andrew Savinov, Jacob A. Hambalek, Anshika Gupta, Diego A. Taquiri-Diaz, Yaotian Zhang, Samuel J. Snyder, A. Katherine Hatstat, Angelika Arada, Nam Hyeong Kim, Haoyu Fan, Ethel Tackie-Yarboi, Dylan Boselli, Lee Schnaider, Chang C. Liu, Gene-Wei Li, Denes Hnisz, David M. Sabatini, William F. DeGrado, Jeremy Wohlwend, Gabriele Corso, Regina Barzilay, and Tommi Jaakkola. BoltzGen: Toward universal binder design. *bioRxiv*, 2025. doi:10.1101/2025.11.20.689494. URL https://www.biorxiv.org/content/10.1101/2025.11.20.689494v2.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35:1026–1028, 2017. doi:10.1038/nbt.3988.

Zhihua Tian, Jiale Zhou, Rubo Wang, Yidong Song, Zhenchao Tang, Tianxu Lv, Zhijian Wu, and Yefeng Zheng. PRIME: A pretrained representation-induced model for 3D molecules in de novo binder design. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 12126–12137, 2026. doi:10.1145/3770855.3818887.

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*, 13:176, 2022. doi:10.1038/s41467-021-27838-9.

Fanhao Wang, Yuzhe Wang, Laiyi Feng, Changsheng Zhang, and Luhua Lai. Target-specific de novo peptide binder design with DiffPepBuilder. *Journal of Chemical Information and Modeling*, 64 (24):9135–9149, 2024. doi:10.1021/acs.jcim.4c00975.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi:10.1038/s41586-023-06415-8.

Fang Wu, Zhengyuan Zhou, Shuting Jin, Xiangxiang Zeng, Jure Leskovec, and Jinbo Xu. Surface-based molecular design with multi-modal flow matching. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 3192–3203, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3711896.3737139. URL https://doi.org/10.1145/3711896.3737139.

Jun Zhang, Yangyang Zhou, Tiantian Zhu, and Zexuan Zhu. PepCCD: A contrastive conditioned diffusion framework for target-specific peptide generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28256–28264, 2026. doi:10.1609/aaai.v40i33.40053.

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004. doi:10.1002/prot.20264.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3c4141c12660ad3625eb4ae845e0a6f9-Abstract-Conference.html.

A Additional Method Details

This appendix summarizes notation and information flow, then details reconstruction, target-only conditioning and latent diffusion, and the corresponding training and generation algorithms. Dataset construction and source extraction are described in Appendices B and C; model architecture and optimization settings are collected in Appendix J.

A.1 Notation and information flow

Subscripts h and x distinguish feature and spatial quantities. We use lowercase $\boldsymbol{z}$ for residue-level autoencoder latent variables and uppercase $\boldsymbol{Z}$ for their row-aligned posterior samples. We use $\boldsymbol{U}$ for diffusion-side quantities, including fixed target conditions and time-dependent peptide states.

The coordinate matrices $\boldsymbol{X}^{\alpha}$, source-feature matrices $\boldsymbol{H}$, posterior-sample matrices $\boldsymbol{Z}$, and diffusion-side matrices $\boldsymbol{U}$ have residue-aligned rows. Residue-indexed vectors are written in bold lowercase; other objects, including sequence contexts and parameter arrays, are defined locally. For a matrix $\boldsymbol{A}$ indexed by C, $\boldsymbol{A}[T]$ and $\boldsymbol{A}[P]$ select target and peptide rows, while $\oplus$ assembles these rows in complex order. $\overline{\boldsymbol{H}}_C$ denotes the normalized source-feature reference. Primes in $\boldsymbol{H}'_C$ and $\mathcal{G}'_P$ mark decoder outputs; the tilde in $\widetilde{\boldsymbol{Z}}_{x,C}$ marks the additional spatial perturbation used in autoencoder training. On peptide states $\boldsymbol{U}^{(t)}_{q,P}$, parenthesized superscripts indicate diffusion time; fixed target conditions have no time superscript. The source index m is omitted when a single source is considered.

Table 5: **Core notation for representation adaptation and target-conditioned diffusion.**

Symbol	Meaning
$T,\ P,\ C$	Retained target, peptide, and complex residue sets, with $C = T \dot{\cup} P$.
$\mathcal{G}_T,\ \mathcal{G}_P$	Supplied target graph and observed training-peptide graph, respectively.
$\boldsymbol{S}_T,\ \boldsymbol{S}_C$	Target-only and complete target–peptide sequence contexts used for source-feature extraction.
$\boldsymbol{H}^{(m)}_T,\ \boldsymbol{H}^{(m)}_C$	Source features from the fixed interface $\mathcal{F}_m$, aligned to target and complex rows; their channel width is d_m.
$\boldsymbol{X}^{\alpha}_T,\ \boldsymbol{X}^{\alpha}_P,\ \boldsymbol{X}^{\alpha}_C$	Observed Cα coordinates in the physical frame, with $\boldsymbol{X}^{\alpha}_C = \boldsymbol{X}^{\alpha}_T \oplus \boldsymbol{X}^{\alpha}_P$; coordinates are expressed in Å.
$\boldsymbol{\mu}_{h,C},\ \boldsymbol{\sigma}_{h,C},\ \boldsymbol{\sigma}_{x,C}$	Encoder-produced feature posterior means and standard deviations, and isotropic spatial posterior standard deviations. Spatial posterior means are the observed coordinates $\boldsymbol{X}^{\alpha}_C$.
$\boldsymbol{Z}_{h,C},\ \boldsymbol{Z}_{x,C}$	Reparameterized autoencoder samples of shape $\lvert C\rvert \times d_h$ and $\lvert C\rvert \times 3$; spatial samples are in physical coordinates.
$\widetilde{\boldsymbol{Z}}_{x,C}$	Spatial posterior samples after the additional training-time perturbation on peptide rows.
$\overline{\boldsymbol{H}}_C,\ \boldsymbol{H}'_C$	Normalized source-feature reference and reconstructed source features.
$\mathcal{G}'_P$	Decoder-produced peptide sequence and bound conformation, in either reconstruction or generation.
$\Phi_T,\ \Phi_T^{-1}$	Target-centered coordinate normalization and its inverse; the center is $\boldsymbol{c}_T$ and the scale is $\kappa_x = 10\,\text{Å}$.
$\boldsymbol{U}_{h,T},\ \boldsymbol{U}_{x,T}$	Fixed target feature condition from $\mathcal{A}_\eta(\boldsymbol{H}_T)$ and spatial condition from $\Phi_T(\boldsymbol{X}^{\alpha}_T)$.
$\boldsymbol{\mu}_{h,C}[T]$	Complete-context target feature means used to supervise the adapter through $\mathcal{L}_{\text{align}}$.
$\boldsymbol{U}^{(0)}_{h,P},\ \boldsymbol{U}^{(0)}_{x,P}$	Clean peptide diffusion states; during training, they are constructed from $\boldsymbol{\mu}_{h,C}[P]$ and $\Phi_T(\boldsymbol{X}^{\alpha}_P)$, respectively.
$\boldsymbol{U}^{(t)}_{q,P},\ q \in \{h, x\}$	Peptide diffusion state of field q at time $t \in \{0, \ldots, N_{\text{diff}}\}$.
$\mathcal{U}_T,\ \mathcal{U}^{(t)}_P$	Pairs of target conditions and peptide diffusion states: $(\boldsymbol{U}_{h,T}, \boldsymbol{U}_{x,T})$ and $(\boldsymbol{U}^{(t)}_{h,P}, \boldsymbol{U}^{(t)}_{x,P})$.

Reconstruction. The encoder maps complete-context source features $\boldsymbol{H}_C$ to posterior parameters. During training, samples $\boldsymbol{Z}_{h,C}$ and $\boldsymbol{Z}_{x,C}$ support reconstruction through $\boldsymbol{H}'_C = \mathcal{D}^{\mathrm{rep}}_\xi(\boldsymbol{Z}_{h,C})$ and $\mathcal{G}'_P = \mathcal{D}^{\mathrm{mol}}_\psi(\boldsymbol{H}'_C, \widetilde{\boldsymbol{Z}}_{x,C}; \mathcal{G}_T)$. The deterministic reconstruction evaluation uses posterior means in place of these random posterior draws, as specified in Appendix H.

Diffusion training. After freezing the autoencoder, the complete-context feature means serve two roles: target rows $\boldsymbol{\mu}_{h,C}[T]$ supervise the adapter, and peptide rows $\boldsymbol{\mu}_{h,C}[P]$ supply the clean feature states. For a training complex, the target-only conditions and clean peptide states are

$$\boldsymbol{U}_{h,T} = \mathcal{A}_\eta(\boldsymbol{H}_T), \quad \boldsymbol{U}_{x,T} = \Phi_T(\boldsymbol{X}^\alpha_T),$$
$$\boldsymbol{U}^{(0)}_{h,P} = \boldsymbol{\mu}_{h,C}[P], \quad \boldsymbol{U}^{(0)}_{x,P} = \Phi_T(\boldsymbol{X}^\alpha_P).$$

The clean peptide feature state $\boldsymbol{U}^{(0)}_{h,P}$ is set directly to the complete-context posterior mean $\boldsymbol{\mu}_{h,C}[P]$, without an additional feature-space transformation. Spatial diffusion states are expressed in the target-centered, scaled coordinate system defined by Φ_T:

$$\Phi_T(\boldsymbol{X}) = \frac{\boldsymbol{X} - \mathbf{1}\boldsymbol{c}_T^\top}{\kappa_x}, \qquad \Phi_T^{-1}(\boldsymbol{Y}) = \kappa_x \boldsymbol{Y} + \mathbf{1}\boldsymbol{c}_T^\top.$$

Here $\boldsymbol{Y}$ is a dummy coordinate argument. The clean spatial state is therefore constructed from the spatial posterior mean, rather than from a sampled $\boldsymbol{Z}_{x,C}$. Only peptide states receive diffusion noise; $\mathcal{U}_T$ remains fixed across diffusion steps. For denoising, $\boldsymbol{U}_{h,T}$ is passed through stop-gradient, so $\mathcal{L}_{\mathrm{diff}}$ does not update the adapter; the adapter is trained through $\mathcal{L}_{\mathrm{align}}$ against $\boldsymbol{\mu}_{h,C}[T]$.

Generation and decoding. Generation constructs $\mathcal{U}_T$ from target-only inputs and initializes the peptide states from Gaussian noise at $t = N_{\mathrm{diff}}$. During generation, reverse diffusion produces $\boldsymbol{U}^{(0)}_{q,P}$ under these target-only conditions. Decoding then proceeds as

$$\boldsymbol{H}'_C = \mathcal{D}^{\mathrm{rep}}_\xi(\boldsymbol{U}_{h,T} \oplus \boldsymbol{U}^{(0)}_{h,P}),$$
$$\mathcal{G}'_P = \mathcal{D}^{\mathrm{mol}}_\psi\big(\boldsymbol{H}'_C, \boldsymbol{X}^\alpha_T \oplus \Phi_T^{-1}(\boldsymbol{U}^{(0)}_{x,P}); \mathcal{G}_T\big).$$

The target feature condition $\boldsymbol{U}_{h,T}$ and generated peptide feature state $\boldsymbol{U}^{(0)}_{h,P}$ already lie in the feature-latent coordinate system expected by the representation decoder, whereas the peptide spatial state is mapped back to physical coordinates by Φ_T^{-1} before peptide decoding. The supplied target graph remains fixed.

Random draws and time variables. Table 6 distinguishes the Gaussian draws used below. Spatial posterior sampling and peptide spatial augmentation are successive operations: augmentation is added to the sampled spatial latent (Equation (19)). Atom-flow time $\tau \in [0, 1]$ is distinct from latent-diffusion time t. The initial peptide state $\boldsymbol{U}^{(N_{\mathrm{diff}})}_{q,P}$ is also sampled from a standard Gaussian, without a separate noise symbol.

A.2 Representation and peptide reconstruction

This section details the posterior distributions, decoding procedure, and reconstruction objectives introduced in Section 3.3. Here, representation reconstruction is implemented as normalized source-feature reconstruction.

Reconstruction scopes. Source-feature reconstruction supervises all complex rows C, whereas residue-type and atom-flow supervision in the peptide decoder cover all peptide residues and a sampled subset of target residues. Let $T_{\mathrm{aux}} \subseteq T$ denote this subset and define

$$M = P \cup T_{\mathrm{aux}}, \qquad \mathcal{I}_R = \{a : i(a) \in R\}, \tag{13}$$

where R is a residue subset and $i(a)$ is the parent residue of atom a. Residue-type and atom-flow losses use M and $\mathcal{I}_M$, respectively. During training, the decoder also predicts residues in T_{aux} for auxiliary supervision, while atoms outside $\mathcal{I}_M$ retain their observed coordinates as fixed context. The notation $\mathcal{G}'_P$ in the main text denotes the peptide output of interest; target predictions on T_{aux} provide auxiliary supervision only. For each training complex, we draw a value from $\mathcal{N}(0, 0.05^2)$ truncated to $[-1, 1]$ and independently include each target residue in T_{aux} with probability equal to the draw's absolute value.

Table 6: **Gaussian draws in reconstruction and diffusion.** Draws are standard normal unless set to zero at the final reverse step. Indices i and a denote residues and atoms; $q \in \{h, x\}$ denotes the feature or spatial field.

Draw	Where it enters	Use
$\boldsymbol{\epsilon}_{h,i}$	Feature posterior, scaled by $\boldsymbol{\sigma}_{h,i}$	Autoencoder reparameterization.
$\boldsymbol{\epsilon}_{x,i}$	Spatial posterior, scaled by $\sigma_{x,i}$	Autoencoder reparameterization around observed Cα coordinates.
$\boldsymbol{\eta}_i$	Additional displacement $\sigma_{\mathrm{aug}}\boldsymbol{\eta}_i$ on peptide spatial samples	Spatial augmentation during autoencoder training only.
$\boldsymbol{\eta}_a$	Atom-prior displacement $\sigma_{\mathrm{atom}}\boldsymbol{\eta}_a$ around the parent residue's spatial input	Atom-flow training; generation uses Gaussian atom initialization at the same scale.
$\boldsymbol{\epsilon}_{q,t}$	Forward corruption of peptide states	Diffusion training; the denoiser predicts this draw.
$\boldsymbol{\xi}_{q,t}$	Reverse-step displacement $\sigma_t^{\mathrm{rev}}\boldsymbol{\xi}_{q,t}$	Generation; sampled for $t > 1$ and set to zero for $t = 1$.

A.2.1 Posterior parameterization and KL

From pre-activations $\boldsymbol{a}_{h,C}$ and $\boldsymbol{a}_{x,C}$, the encoder in Equation (2) parameterizes log variances for feature latents and spatial latents as

$$\boldsymbol{\ell}_{h,C} = -|\boldsymbol{a}_{h,C}|, \qquad \boldsymbol{\ell}_{x,C} = -|\boldsymbol{a}_{x,C}|, \qquad \boldsymbol{\sigma}_{q,C} = \exp(\boldsymbol{\ell}_{q,C}/2), \quad q \in \{h, x\}. \tag{14}$$

Each residue has d_h standard deviations for the feature latent and one isotropic spatial standard deviation; independent Gaussian draws yield

$$\begin{aligned} \boldsymbol{z}_{h,i} &= \boldsymbol{\mu}_{h,i} + \boldsymbol{\sigma}_{h,i} \odot \boldsymbol{\epsilon}_{h,i}, \quad & \boldsymbol{\epsilon}_{h,i} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_h}), \\ \boldsymbol{z}_{x,i} &= \boldsymbol{x}_i^{\alpha} + \sigma_{x,i}\,\boldsymbol{\epsilon}_{x,i}, & \boldsymbol{\epsilon}_{x,i} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_3). \end{aligned} \tag{15}$$

Stacking the residue-level realizations in complex row order gives the posterior sample matrices $\boldsymbol{Z}_{h,C}$ and $\boldsymbol{Z}_{x,C}$ used in Equation (4). KL regularization over the active set M uses $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_h})$ as the prior for feature latents and $\mathcal{N}(\boldsymbol{x}_i^{\alpha}, \boldsymbol{I}_3)$ as the spatial reference. After averaging over residues and channels, the losses are

$$\begin{aligned} \mathcal{L}_{\mathrm{KL}}^{h} &= \frac{1}{2|M|d_h} \sum_{i \in M} \sum_{j=1}^{d_h} \left[\mu_{h,ij}^2 + e^{\ell_{h,ij}} - \ell_{h,ij} - 1\right], \\ \mathcal{L}_{\mathrm{KL}}^{x} &= \frac{1}{2|M|} \sum_{i \in M} \left[e^{\ell_{x,i}} - \ell_{x,i} - 1\right]. \end{aligned} \tag{16}$$

Because the spatial posterior and reference share the observed mean, their KL has no mean-displacement term; averaging three identical coordinate-channel terms yields the second expression.

A.2.2 Normalized source-feature reconstruction

Feature latents pass through an up-projection to width d_m, a Transformer, and output LayerNorm to produce reconstructed source features $\boldsymbol{H}'_C$. Reconstruction uses the normalized source-feature reference $\overline{\boldsymbol{H}}_C$:

$$\boldsymbol{H}'_C = \mathrm{LN}_{\mathrm{out}}\left[\mathcal{T}_{\xi}^{\mathrm{rep}}(\boldsymbol{Z}_{h,C}\boldsymbol{W}_{\mathrm{up}} + \mathbf{1}\boldsymbol{b}_{\mathrm{up}}^{\top})\right], \qquad \overline{\boldsymbol{H}}_C = \mathrm{LN}_0(\boldsymbol{H}_C). \tag{17}$$

Here $\boldsymbol{W}_{\mathrm{up}} \in \mathbb{R}^{d_h \times d_m}$, and $\mathcal{T}_{\xi}^{\mathrm{rep}}$ denotes the Transformer. Row-wise LayerNorm uses learned affine parameters for the decoder output ($\mathrm{LN}_{\mathrm{out}}$) and none for the reference (LN_0). With $\boldsymbol{h}'_i$ and $\overline{\boldsymbol{h}}_i$ denoting corresponding rows of the reconstructed and reference matrices, source-feature supervision combines

Smooth-L1 with unit transition (SL1) and row-wise cosine losses:

$$\begin{aligned}\mathcal{L}_{\mathrm{sl1}} &= \frac{1}{|C| d_m} \sum_{i \in C} \sum_{j=1}^{d_m} \mathrm{SL1}\big(h'_{ij} - \overline{h}_{ij}\big)\,, \\ \mathcal{L}_{\cos} &= \frac{1}{|C|} \sum_{i \in C} \big[1 - \cos\big(\boldsymbol{h}'_i, \overline{\boldsymbol{h}}_i\big)\big]\,, \\ \mathcal{L}_{\mathrm{rep}} &= 0.5\,\mathcal{L}_{\mathrm{sl1}} + 2\,\mathcal{L}_{\cos}.\end{aligned} \tag{18}$$

Here $\cos$ denotes row-wise cosine similarity.

A.2.3 Peptide reconstruction

Peptide decoding uses reconstructed source features $\boldsymbol{H}'_C$, residue-level spatial inputs, and the supplied target graph $\mathcal{G}_T$ to reconstruct residue identities and bound geometry. At generation time, the predicted identities define atom templates whose coordinates are refined by the structure module; atom features from its final call then support inter-residue bond prediction.

Spatial augmentation, identities, and topology. During autoencoder training, peptide spatial latents receive an additional Gaussian perturbation, denoted by a tilde:

$$\widetilde{\boldsymbol{z}}_{x,i} = \boldsymbol{z}_{x,i} + \mathbb{I}_{\{i \in P\}} \sigma_{\mathrm{aug}} \boldsymbol{\eta}_i, \qquad \boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_3), \qquad \widetilde{\boldsymbol{Z}}_{x,C} = \{\widetilde{\boldsymbol{z}}_{x,i}\}_{i \in C}. \tag{19}$$

With $\sigma_{\mathrm{aug}} = 1$ Å in the reported setting, this perturbation is separate from posterior sampling and is omitted during generation. During autoencoder training, target spatial rows remain posterior samples centered at the observed target Cα coordinates, whereas only peptide rows receive the additional perturbation. Residue-type prediction and atom initialization both use these spatial rows. Identity supervision compares each predicted distribution $\boldsymbol{\pi}_i$ with the observed residue identity s_i:

$$\mathcal{L}_{\mathrm{type}} = -\frac{1}{|M|} \sum_{i \in M} \log \pi_i(s_i). \tag{20}$$

Coordinate reconstruction during training uses observed atom types and topology $\mathcal{B}^{\mathrm{known}}$, including target–target and intra-residue bonds. Peptide-involving inter-residue bonds are supplied with probability 0.5 per complex during training; during generation they are not supplied and are instead predicted from the final atom features.

Atom flow. For each complex, atom-flow training samples a single $\tau \sim \mathrm{Unif}[0, 1]$; this continuous flow time is distinct from the latent-diffusion time t. An active atom's prior is centered on the augmented spatial latent of its parent residue:

$$\begin{aligned}\boldsymbol{x}_a^{\mathrm{pri}} &= \begin{cases} \widetilde{\boldsymbol{z}}_{x,i(a)} + \sigma_{\mathrm{atom}} \boldsymbol{\eta}_a, & a \in \mathcal{I}_M, \\ \boldsymbol{x}_a^{\mathrm{obs}}, & a \notin \mathcal{I}_M, \end{cases} \\ \boldsymbol{X}_\tau &= (1 - \tau)\boldsymbol{X}^{\mathrm{obs}} + \tau \boldsymbol{X}^{\mathrm{pri}}, \qquad\qquad \boldsymbol{V}^{\mathrm{target}} = \boldsymbol{X}^{\mathrm{obs}} - \boldsymbol{X}^{\mathrm{pri}}.\end{aligned} \tag{21}$$

Thus, $\tau = 1$ corresponds to the atom prior and $\tau = 0$ to the observed coordinates. Here $\boldsymbol{X}^{\mathrm{obs}}$ contains the observed atomic coordinates, $\boldsymbol{\eta}_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_3)$, and $\sigma_{\mathrm{atom}} = 1$ Å is the atom-prior standard deviation. Conditioned on atom types, supplied topology, and reconstructed source features $\boldsymbol{H}'_C$, the structure module predicts $\boldsymbol{X}'_{\mathrm{next}}$ from $\boldsymbol{X}_\tau$ and τ. Subtracting $\boldsymbol{X}_\tau$ from this output gives the predicted displacement, which is used to estimate the endpoint:

$$\boldsymbol{V}'_\tau = \boldsymbol{X}'_{\mathrm{next}} - \boldsymbol{X}_\tau, \qquad \boldsymbol{X}'_{\mathrm{end}} = \boldsymbol{X}_\tau + \tau \boldsymbol{V}'_\tau. \tag{22}$$

Displacement supervision sums the three coordinate errors before averaging over active atoms:

$$\mathcal{L}_{\mathrm{vec}} = \frac{1}{|\mathcal{I}_M|} \sum_{a \in \mathcal{I}_M} \|\boldsymbol{v}'_{\tau,a} - \boldsymbol{v}_a^{\mathrm{target}}\|_2^2. \tag{23}$$

Geometric losses use the endpoint estimate, as detailed below.

Coordinate integration during generation. After restoring the final peptide spatial states $\boldsymbol{U}_{x,P}^{(0)}$ to physical coordinates, we initialize each peptide atom around its parent residue's restored position with Gaussian noise of standard deviation σ_{atom}. Target atoms remain fixed. The observed target Cα coordinates and restored peptide coordinates provide spatial anchors, whereas $\mathcal{G}_T$ supplies the fixed target atom graph, topology, and coordinates used by the peptide decoder. Let $1 = \tau_0 > \cdots > \tau_K = 0$ and $h_k = \tau_k - \tau_{k+1}$. For atom types $\boldsymbol{A}$, the flow update is

$$\begin{aligned}(\boldsymbol{F}'^{(k)}, \boldsymbol{X}'^{(k)}_{\text{next}}) &= \mathcal{D}^{\text{str}}_{\psi}\big(\boldsymbol{A}, \mathcal{B}^{\text{known}}, \boldsymbol{X}^{(k)}, \tau_k; \boldsymbol{H}'_C\big),\\ \boldsymbol{V}'_{\tau_k} &= \boldsymbol{X}'^{(k)}_{\text{next}} - \boldsymbol{X}^{(k)},\\ \boldsymbol{X}^{(k+1)}_{\mathcal{I}_P} &= \boldsymbol{X}^{(k)}_{\mathcal{I}_P} + h_k \boldsymbol{V}'_{\tau_k,\mathcal{I}_P}, \qquad \boldsymbol{X}^{(k+1)}_{\mathcal{I}_T} = \boldsymbol{X}^{\text{obs}}_{\mathcal{I}_T}.\end{aligned} \tag{24}$$

Here $\mathcal{D}^{\text{str}}_{\psi}$ is the structure module within the peptide decoder $\mathcal{D}^{\text{mol}}_{\psi}$, and $\boldsymbol{F}'^{(k)}$ denotes its atom-feature output. A uniform schedule sets $\tau_k = 1 - k/K$ with $K = 10$ in the reported setting. Bond prediction uses atom features from the final structure-module call and neighborhoods computed from the final coordinates.

Bond and geometric supervision. Inter-residue bond prediction uses cross-entropy $\mathcal{L}_{\text{bond}}$ over labeled atom pairs $\mathcal{Q}$, including a no-bond class. Geometric supervision is applied to the endpoint estimate $\boldsymbol{X}'_{\text{end}}$: $\mathcal{L}_{\text{blen}}$ penalizes peptide bond-length errors, $\mathcal{L}_{\text{local}}$ penalizes local atom-pair distance errors for pairs within $6\,\text{Å}$ in the observed structure, and $\mathcal{L}_{\text{chain}}$ penalizes squared deviations of consecutive Cα distances from $3.8\,\text{Å}$.

Loss weights. The weighted sum of identity, displacement, bond, and geometric losses defines $\mathcal{L}_{\text{mol}}$, which enters the autoencoder objective in Equation (6) alongside source-feature reconstruction and KL regularization. Table 7 reports the shared loss weights and reduction domains for the matched three-source experiments. Linear warm-up over $4{,}000$ optimizer updates raises the feature and spatial KL weights to 0.4 and 0.6, respectively. All reductions pool the selected entries across the minibatch.

Table 7: Autoencoder loss weights and reduction domains for the matched three-source experiments. KL weights are the values after warm-up.

Term	Weight	Averaging domain
$\mathcal{L}_{\text{sl1}}$	0.5	Complex rows and source-feature channels
$\mathcal{L}_{\cos}$	2.0	Complex rows
$\mathcal{L}_{\text{type}}$	1.0	Active residues M
$\mathcal{L}_{\text{vec}}$	2.0	Active atoms $\mathcal{I}_M$; squared errors summed over coordinates
$\mathcal{L}_{\text{bond}}$	1.0	Entries of the labeled pair list $\mathcal{Q}$
$\mathcal{L}_{\text{blen}}$	0.2	Bonds with two peptide endpoints, $\tau < 0.25$
$\mathcal{L}_{\text{local}}$	0.5	Selected local pairs with an active endpoint, $\tau < 0.25$
$\mathcal{L}_{\text{chain}}$	0.6	Consecutive same-chain residue pairs with both endpoints in M
$\mathcal{L}^{h}_{\text{KL}}$	0.4	Active residues M and feature-latent channels
$\mathcal{L}^{x}_{\text{KL}}$	0.6	Active residues M and three spatial channels

A.3 Target conditioning and latent diffusion

This section specifies target-only conditioning, the denoising objective, and reverse sampling for peptide generation.

Target-only conditioning. During both diffusion training and generation, conditioning is constructed from target-only source features $\boldsymbol{H}_T$ and the supplied target geometry. The adapter produces the target feature condition

$$\boldsymbol{U}_{h,T} = \mathcal{A}_{\eta}(\boldsymbol{H}_T). \tag{25}$$

Together with $\boldsymbol{U}_{x,T} = \Phi_T(\boldsymbol{X}^{\alpha}_T)$, these outputs form the fixed target condition $\mathcal{U}_T = (\boldsymbol{U}_{h,T}, \boldsymbol{U}_{x,T})$.

Known complexes provide alignment supervision for the adapter. Encoding the complete-context source features $\boldsymbol{H}_C$ with the frozen encoder yields target posterior means $\boldsymbol{\mu}_{h,C}[T]$, which serve as

alignment targets:

$$\mathcal{L}_{\text{align}} = 0.3\,\frac{\|\boldsymbol{U}_{h,T} - \boldsymbol{\mu}_{h,C}[T]\|_F^2}{|T|d_h} + 0.1\left[1 - \frac{1}{|T|}\sum_{i\in T}\cos\big((\boldsymbol{U}_{h,T})_i, (\boldsymbol{\mu}_{h,C}[T])_i\big)\right]. \tag{26}$$

Here cos denotes cosine similarity. The complete-context target means are used only as alignment targets; denoising and generation use $\boldsymbol{U}_{h,T}$. For denoising, $\boldsymbol{U}_{h,T}$ is passed through stop-gradient, so $\mathcal{L}_{\text{diff}}$ does not update the adapter; the adapter is trained through $\mathcal{L}_{\text{align}}$.

Forward diffusion and training objective. For each training complex, the frozen encoder's peptide posterior means $\boldsymbol{\mu}_{h,C}[P]$ directly define the clean feature states $\boldsymbol{U}_{h,P}^{(0)}$, while observed peptide Cα coordinates transformed by Φ_T define the clean spatial states $\boldsymbol{U}_{x,P}^{(0)}$, as in Equation (9). Forward corruption follows Equation (10). Only peptide states receive diffusion noise; $\mathcal{U}_T$ remains fixed.

During training, each complex independently samples a diffusion timestep and independent Gaussian noise for the two peptide fields. The equivariant denoiser (Kong et al., 2025) takes the peptide states, the fixed target condition $\mathcal{U}_T$, and an embedding of the schedule coefficient β_t defined below. It predicts the sampled Gaussian targets for both fields. We write $\widehat{\boldsymbol{\epsilon}}_{q,t}$ for the prediction in field q and $\boldsymbol{\epsilon}_{q,t}$ for its sampled target.

Let $\mathcal{P}_{\text{mb}}$ collect peptide residue rows in a minibatch and $N_P = |\mathcal{P}_{\text{mb}}|$. Each row uses the diffusion time sampled for its complex. For each field $q \in \{h, x\}$, the loss sums over channels and averages over peptide rows:

$$\mathcal{L}_q = \frac{1}{N_P}\sum_{i\in\mathcal{P}_{\text{mb}}} \|\widehat{\boldsymbol{\epsilon}}_{q,t,i} - \boldsymbol{\epsilon}_{q,t,i}\|_2^2,$$
$$\mathcal{L}_{\text{diff}} = \mathcal{L}_x + \frac{3}{d_h}\mathcal{L}_h. \tag{27}$$

The factor $3/d_h$ balances the feature- and spatial-field losses according to their channel counts. Combining denoising and alignment gives Equation (11), with $\lambda_{\text{align}} = 1$.

Reverse sampling and decoding. Generation initializes both peptide diffusion states from independent Gaussian noise at $t = N_{\text{diff}}$ and applies reverse updates for $t = N_{\text{diff}}, \ldots, 1$, while keeping $\mathcal{U}_T$ fixed. Here N_{diff} is the number of reverse-diffusion steps. We use the cosine noise schedule

$$g_t = \cos^2\left[\frac{\pi}{2}\frac{t/N_{\text{diff}} + s}{1+s}\right], \quad \overline{a}_t = \frac{g_t}{g_0},$$
$$\alpha_t = \sqrt{\overline{a}_t}, \qquad \sigma_t = \sqrt{1-\overline{a}_t}. \tag{28}$$

We use $s = 0.01$ and $N_{\text{diff}} = 100$. Here $\overline{a}_t$ is the marginal coefficient. The reverse sampler uses the step coefficients

$$\beta_t = \min\left(1 - \frac{\overline{a}_t}{\overline{a}_{t-1}}, 0.999\right), \qquad a_t = 1 - \beta_t, \qquad t = 1, \ldots, N_{\text{diff}}. \tag{29}$$

Noise in each reverse step has variance and standard deviation

$$\beta_t^{\text{rev}} = \frac{1-\overline{a}_{t-1}}{1-\overline{a}_t}\beta_t, \qquad \sigma_t^{\text{rev}} = \sqrt{\beta_t^{\text{rev}}}, \qquad t = 1, \ldots, N_{\text{diff}}. \tag{30}$$

Following the UniMoMo implementation (Kong et al., 2025), the reverse mean uses a lower-bounded step coefficient:

$$\gamma_t = \max(a_t, a_{N_{\text{diff}}-1}), \qquad t = 1, \ldots, N_{\text{diff}}. \tag{31}$$

For the cosine schedule, this changes only the initial reverse step $N_{\text{diff}} \to N_{\text{diff}} - 1$, limiting the large multiplier $1/\sqrt{a_t}$ caused by the small terminal coefficient. For each field $q \in \{h, x\}$,

$$\boldsymbol{U}_{q,P}^{(t-1)} = \frac{1}{\sqrt{\gamma_t}}\left(\boldsymbol{U}_{q,P}^{(t)} - \frac{1-\gamma_t}{\sqrt{1-\overline{a}_t}}\widehat{\boldsymbol{\epsilon}}_{q,t}\right) + \sigma_t^{\text{rev}}\boldsymbol{\xi}_{q,t}, \tag{32}$$

where $\boldsymbol{\xi}_{q,t} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for $t > 1$ and $\boldsymbol{\xi}_{q,1} = \mathbf{0}$; the feature and spatial fields use independent reverse noises. For numerical stability, 10^{-8} is added inside each denominator square root.

During generation, reverse diffusion produces $\boldsymbol{U}_{q,P}^{(0)}$. The generated feature states $\boldsymbol{U}_{h,P}^{(0)}$ are assembled with $\boldsymbol{U}_{h,T}$ and passed directly through the representation decoder to produce reconstructed source features $\boldsymbol{H}'_C$. The peptide spatial states $\boldsymbol{U}_{x,P}^{(0)}$ are restored to physical coordinates by Φ_T^{-1} and assembled with the supplied target Cα coordinates. The peptide decoder combines these spatial inputs with $\boldsymbol{H}'_C$ and the supplied target graph to produce $\mathcal{G}'_P$, as in Equation (12). Algorithm 3 summarizes generation.

A.4 Training and generation algorithms

Algorithms 1–3 summarize autoencoder training, latent diffusion training, and peptide generation. Appendices A.2 and A.3 specify the reconstruction losses and diffusion updates.

Algorithm 1 Train the autoencoder

Input: Training complexes and fixed source interface $\mathcal{F}_m$
Output: Trained autoencoder parameters (ϕ, ξ, ψ)

1: **for** each optimizer update r **do**
2: Extract $\boldsymbol{H}_C$ and set $\overline{\boldsymbol{H}}_C \leftarrow \mathrm{LN}_0(\boldsymbol{H}_C)$.
3: Sample $T_{\mathrm{aux}} \subseteq T$ and form $M = P \cup T_{\mathrm{aux}}$ and $\mathcal{I}_M$.
4: Encode $\boldsymbol{H}_C$ and draw posterior samples $\boldsymbol{Z}_{h,C}$ and $\boldsymbol{Z}_{x,C}$ for the feature and spatial latents, using observed Cα coordinates as the spatial posterior means.
5: Decode $\boldsymbol{H}'_C \leftarrow \mathcal{D}_\xi^{\mathrm{rep}}(\boldsymbol{Z}_{h,C})$ and compute $\mathcal{L}_{\mathrm{rep}}$.
6: Perturb peptide spatial rows to form $\widetilde{\boldsymbol{Z}}_{x,C}$ and predict residue identities on M using $\boldsymbol{H}'_C$, $\widetilde{\boldsymbol{Z}}_{x,C}$, and the supplied target graph.
7: Reconstruct atomic coordinates using observed atom types and the training-time supplied topology.
8: Predict inter-residue bonds from the final atom features and coordinate neighborhoods.
9: Compute $\mathcal{L}_{\mathrm{mol}}$ and both KL terms (Appendix A.2).
10: Update (ϕ, ξ, ψ) with $\mathcal{L}_{\mathrm{AE}}(r)$.
11: **end for**

Algorithm 2 Train the diffusion model and target adapter

Input: Training complexes, fixed source interface, and frozen autoencoder
Output: Trained diffusion parameters θ and adapter parameters η

1: **for** each optimizer update **do**
2: Extract $\boldsymbol{H}_C$ and independently compute target-only $\boldsymbol{H}_T$.
3: Encode $\boldsymbol{H}_C$ with the frozen encoder to obtain $\boldsymbol{\mu}_{h,C}$; set $\boldsymbol{U}_{h,P}^{(0)} \leftarrow \boldsymbol{\mu}_{h,C}[P]$ and $\boldsymbol{U}_{x,P}^{(0)} \leftarrow \Phi_T(\boldsymbol{X}_P^\alpha)$.
4: Compute $\boldsymbol{U}_{h,T} \leftarrow \mathcal{A}_\eta(\boldsymbol{H}_T)$ and $\mathcal{L}_{\mathrm{align}}$ against $\boldsymbol{\mu}_{h,C}[T]$.
5: Set $\boldsymbol{U}_{x,T} \leftarrow \Phi_T(\boldsymbol{X}_T^\alpha)$ and form $\mathcal{U}_T = (\boldsymbol{U}_{h,T}, \boldsymbol{U}_{x,T})$, using a stop-gradient copy of $\boldsymbol{U}_{h,T}$ only for denoising.
6: Sample t and independent Gaussian noise; form $\boldsymbol{U}_{q,P}^{(t)}$ for $q \in \{h, x\}$ using Equation (10).
7: Predict $(\widehat{\boldsymbol{\epsilon}}_{h,t}, \widehat{\boldsymbol{\epsilon}}_{x,t})$ from $\mathcal{U}_P^{(t)}$ under the fixed target-side conditioning.
8: Update (θ, η) with $\mathcal{L}_{\mathrm{LDM}}$; the stop-gradient makes $\mathcal{L}_{\mathrm{diff}}$ update θ and $\mathcal{L}_{\mathrm{align}}$ update η.
9: **end for**

Algorithm 3 Generate a peptide from a protein target

Input: Target sequence context $\boldsymbol{S}_T$, target graph $\mathcal{G}_T$, peptide length L, fixed source interface $\mathcal{F}_m$, and trained parameters $(\xi, \psi, \eta, \theta)$

Output: Peptide sequence and bound conformation in $\mathcal{G}'_P$

1: Instantiate peptide rows P with $|P| = L$ and extract $\boldsymbol{H}_T \leftarrow \mathcal{F}_m(\boldsymbol{S}_T; T)$.
2: Compute $\boldsymbol{U}_{h,T} \leftarrow \mathcal{A}_\eta(\boldsymbol{H}_T)$ and $\boldsymbol{U}_{x,T} \leftarrow \Phi_T(\boldsymbol{X}_T^\alpha)$; form the fixed target condition $\mathcal{U}_T = (\boldsymbol{U}_{h,T}, \boldsymbol{U}_{x,T})$.
3: Initialize $\boldsymbol{U}_{q,P}^{(N_{\text{diff}})} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ independently for $q \in \{h, x\}$.
4: **for** $t = N_{\text{diff}}, \ldots, 1$ **do**
5: Predict $(\widehat{\boldsymbol{\epsilon}}_{h,t}, \widehat{\boldsymbol{\epsilon}}_{x,t})$ from $\mathcal{U}_P^{(t)}$ under the fixed target-side conditioning.
6: Update the peptide states using Equation (32).
7: **end for**
8: Decode $\boldsymbol{H}'_C \leftarrow \mathcal{D}_\xi^{\text{rep}}(\boldsymbol{U}_{h,T} \oplus \boldsymbol{U}_{h,P}^{(0)})$.
9: Restore the generated peptide spatial state $\boldsymbol{U}_{x,P}^{(0)}$ by Φ_T^{-1} and decode $\mathcal{G}'_P$ from $\boldsymbol{H}'_C$, $\boldsymbol{X}_T^\alpha \oplus \Phi_T^{-1}(\boldsymbol{U}_{x,P}^{(0)})$, and $\mathcal{G}_T$: predict peptide identities, instantiate atom templates, integrate coordinates, and predict inter-residue bonds (Appendix A.2.3).
10: **return** $\mathcal{G}'_P$

B Dataset Construction and Split

The training and validation pool is constructed from target–peptide complexes in PepMerge (Li et al., 2024) and the supervised target–peptide subset of PepBench (Kong et al., 2024). Records from the two sources are converted to a common target–peptide complex format. Exact merge duplicates are resolved using a complex identifier normalized for letter case and defined by the receptor chains, peptide chain, and PDB or long-form structure identifier. Following PepFlow (Li et al., 2024), peptides are restricted to 3–25 residues. The retained set is further restricted to canonical peptides whose provided FASTA sequences exactly match the sequences recovered from structure. The ProtFrag subset of PepBench is excluded because it consists of intrachain fragments excised from monomeric proteins rather than target–peptide complexes.

For each retained complex, the binding pocket contains target residues whose Cβ-based distance to any peptide residue is below 10 Å, following PepGLAD (Kong et al., 2024). These residues are retained in the target graph $\mathcal{G}_T$ used by the generative model.

The original LNR benchmark (Tsaban et al., 2022) contains 93 targets. Applying the same 3–25-residue peptide-length criterion removes `2HWL`, whose peptide contains 27 residues, leaving 92 evaluation targets. Throughout the paper, LNR denotes this filtered evaluation set.

Following PepFlow (Li et al., 2024), we use target-level clustering and cluster-wise partitioning to construct training and validation splits that are disjoint from LNR. Target sequences from the retained pool and the filtered LNR set are jointly clustered with MMseqs2 (Steinegger & Söding, 2017) using a minimum sequence identity of 40% and a minimum coverage of 80% (`-min-seq-id 0.4 -c 0.8 -cov-mode 0`). Complexes whose targets belong to clusters containing an LNR target are excluded. The remaining 1,938 target clusters are shuffled using NumPy `RandomState(42)`; 193 clusters are assigned to validation and the remaining 1,745 to training. The resulting split contains 1,064 validation and 9,113 training complexes.

Figure 3 summarizes the data-source composition and selected peptide- and target-level statistics of PepDataBench and LNR.

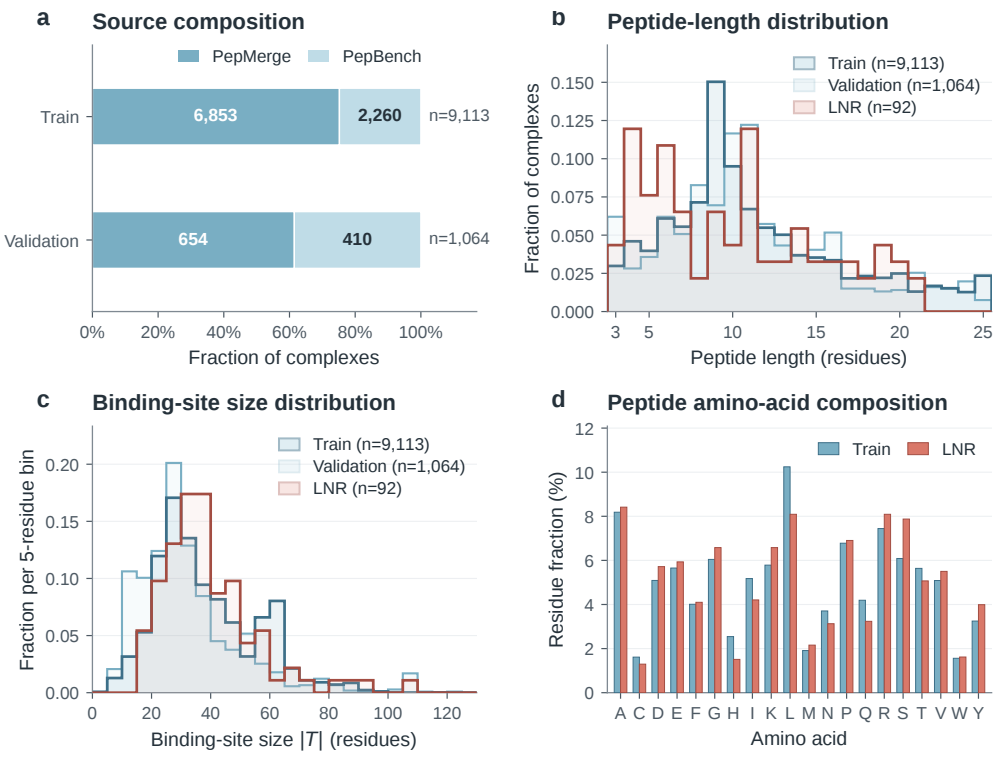


Figure 3: **Dataset statistics for PEPDATABENCH and LNR.** (a) Data-source composition of the training and validation splits after preprocessing. (b) Peptide-length distributions for training, validation, and LNR. (c) Target-pocket size distributions, where $|T|$ is the number of target residues retained in the target graph. (d) Peptide amino-acid composition of the training split and LNR.

C Source-Model Feature Extraction and Preprocessing

C.1 Sequence contexts and model inputs

We use two sequence contexts for source-model feature extraction. The target-only context $\boldsymbol{S}_T$ contains the full target chains, including residues outside the retained pocket, whereas the complete context $\boldsymbol{S}_C$ additionally contains the peptide sequence of the training complex. For $R \in \{T, C\}$, source-model outputs from $\boldsymbol{S}_R$ are mapped to the corresponding graph residues to obtain source features $\boldsymbol{H}_R^{(m)}$. Target-only and complete-context features are extracted separately.

Source-model inputs are constructed only from amino-acid sequences and chain segmentation; observed complex coordinates enter PepRAE separately through the graphs and spatial latents. Complete-context source features are used for autoencoder training and to obtain the complete-context feature means used as clean feature targets during diffusion training. Target-only source features provide the representation-derived target-side conditioning used during diffusion training and generation.

C.2 Fixed source interface and residue mapping

We make the source interface in Equation (1) explicit as

$$\begin{aligned} \mathcal{F}_m(\,\cdot\,; R) &= \Pi_{m,R} \circ \mathrm{State}_m \circ \mathrm{Prep}_m, \\ \boldsymbol{H}_R^{(m)} &= \mathcal{F}_m(\boldsymbol{S}_R; R), \qquad R \in \{T, C\}. \end{aligned} \tag{33}$$

Here Prep_m constructs the inputs for source model m, and State_m runs the model and retrieves its designated residue-level internal representations. The map $\Pi_{m,R}$ selects graph residues in their prescribed order, discards special-token, artificial-linker, and padding positions, and applies any fixed dimensional preprocessing. The resulting source features satisfy $\boldsymbol{H}_R^{(m)} \in \mathbb{R}^{|R| \times d_m}$. The same frozen source-model parameters and preprocessing rules are used for both contexts and remain unchanged during PepRAE training.

Within PepRAE, the encoder processes the target and peptide rows of $\boldsymbol{H}_C$ jointly. The resulting target feature posterior means $\boldsymbol{\mu}_{h,C}[T]$ provide alignment targets for the adapter applied to $\boldsymbol{H}_T$.

C.3 Primary representation sources

The three source interfaces yield features of width $d_m = 384$ after their source-specific preprocessing.

Boltz-2. Boltz-2 jointly processes all chains in each context and provides its final 384-dimensional Pairformer-derived single representation $\boldsymbol{s}_i$ (Passaro et al., 2025). Extraction uses the query sequences without homolog-derived MSA rows, templates, or native-contact constraints. The fixed source interface retains only this single representation; the source-model pair representation is not included. Its native width already matches $d_m = 384$, so no dimensional reduction is applied.

ESM-2. We extract 640-dimensional residue representations from `esm2_t30_150M_UR50D` (Lin et al., 2023). For the complete context, the target chains are encoded as one group and the peptide chain separately; their residue representations are assembled before graph-row selection. Target-only extraction encodes the same target group without the peptide. Consequently, the target rows of the resulting source features are unchanged between the two contexts; complete-context dependence is introduced subsequently when the PepRAE encoder jointly processes the assembled target and peptide rows. A fixed, non-whitened PCA transform maps the 640-dimensional representations to the common source-feature width of 384:

$$\boldsymbol{H}_R^{(\mathrm{ESM2})} = \left(\boldsymbol{H}_{R,\mathrm{raw}}^{(\mathrm{ESM2})} - \mathbf{1}\boldsymbol{\mu}_{\mathrm{PCA}}^{\top}\right) \boldsymbol{W}_{\mathrm{PCA}}, \qquad R \in \{T, C\}. \tag{34}$$

Here $\boldsymbol{H}_{R,\mathrm{raw}}^{(\mathrm{ESM2})}$ contains the unprojected residue representations in graph order, while $\boldsymbol{\mu}_{\mathrm{PCA}}$ and $\boldsymbol{W}_{\mathrm{PCA}}$ are the fitted mean and retained principal directions. PCA is fitted once using all residue representations from the complete sequence contexts of training complexes, including target residues outside the retained pockets. Validation and evaluation data are excluded from this fit. The same frozen transform is applied to complete-context and target-only inputs across all splits before the PepRAE encoder maps the resulting source features to feature latents of width d_h.

LightEnc. LightEnc is a one-layer sequence Transformer with hidden width 384, eight attention heads, and feed-forward width 1536. Amino-acid identity, chain identity, and within-chain position are encoded in its input embeddings, and self-attention operates across all chains in each context. LightEnc is trained from random initialization with masked language modeling (MLM) on the complete sequence contexts $\boldsymbol{S}_C$ of the PEPDATABENCH training complexes. For each training complex, 15% of valid residue positions are selected: 80% of these positions receive a mask token, 10% a randomly sampled residue token, and 10% remain unchanged. Training runs for 100 epochs, with checkpoint selection based on the mean MLM loss over three fixed corrupted versions of the validation sequences. The selected epoch-92 LightEnc checkpoint is frozen before PepRAE training.

D EVALUATION PROTOCOL AND METRICS

We assess sequence and bound-conformation recovery, contact and interface recovery, Rosetta interface scores, Cα clash rates, dihedral-distribution distances, and candidate-pool diversity. This appendix defines the reported generation metrics and their aggregation. Unless stated otherwise, metrics are computed from the generated candidates before relaxation; Rosetta relaxation is used only for ΔG and the derived IMP.

D.1 CANDIDATE-POOL AGGREGATION

Following the candidate-pool evaluation conventions of UniMoMo (Kong et al., 2025) and PepTri (Nguyen et al., 2026), we generate 100 candidates per target in each LNR evaluation run and apply the same metric-specific aggregation rules to PepRAE and all evaluated baselines. Table 8 summarizes the corresponding within-target and run-level reductions. Within each target, maxima and minima are selected independently for each metric, so different columns need not describe the same generated candidate. Dihedral-distribution distances instead pool generated angles across the entire run before comparison with fixed reference profiles.

For configurations evaluated with three diffusion-model training runs, each trained model generates its own pool of 100 candidates per LNR target. We apply the complete aggregation procedure separately to each run and use the mean of the three run-level statistics as the central estimate. Where reported, their sample standard deviation summarizes observed run-to-run variation under a fixed autoencoder and common sampling and evaluation protocol.

D.2 SEQUENCE AND BOUND-CONFORMATION RECOVERY

Amino-acid recovery (AAR). Generated and reference peptide sequences are globally aligned by Needleman–Wunsch (Needleman & Wunsch, 1970) using BLOSUM62 (Henikoff & Henikoff, 1992) in Biopython (Cock et al., 2009). Gap-opening and gap-extension penalties are -10 and -0.5, respectively, with unpenalized terminal gaps. For aligned strings $(a'_\ell)_{\ell=1}^m$ and $(a_\ell)_{\ell=1}^m$, the fraction of identical aligned residues is

$$\mathrm{AAR} = \frac{1}{m}\sum_{\ell=1}^{m} \mathbb{I}[a'_\ell = a_\ell]. \tag{35}$$

Gaps remain in the denominator and do not count as matches. AAR is reported as a percentage.

Complex and ligand RMSD (C-RMSD and L-RMSD). For RMSD evaluation, peptide residues are paired by their benchmark chain and residue identifiers, independently of sequence alignment. For n_α matched Cα pairs, let $\boldsymbol{x}'_i$ and $\boldsymbol{x}_i$ denote generated and reference coordinates.

Complex RMSD (C-RMSD) measures peptide Cα coordinate recovery directly in the fixed target coordinate frame, without peptide-level rigid-body superposition:

$$\text{C-RMSD} = \left[\frac{1}{n_\alpha}\sum_{i=1}^{n_\alpha} \|\boldsymbol{x}'_i - \boldsymbol{x}_i\|_2^2\right]^{1/2}. \tag{36}$$

Table 8: **Aggregation of generation metrics.** The same metric-specific rules are applied to PepRAE and all evaluated baselines.

Metric	Aggregation within an evaluation run
AAR, TM-score, $\mathrm{F1}_{\mathrm{intra}}$, DockQ	Maximum over candidates for each target; mean across targets.
C-, L-, M-RMSD	Minimum over candidates for each target; median across targets.
Rosetta ΔG	Minimum over candidates for each target; mean across targets.
IMP	Whether any candidate satisfies $\Delta G \leq \Delta G^{\mathrm{native}}$ for each target; report the fraction of targets (%).
$\mathrm{Clash}_{\mathrm{in}}$, $\mathrm{Clash}_{\mathrm{out}}$	Mean over candidates for each target; mean across targets.
D_{seq}, D_{pose}, D_{codesign}	Compute the candidate-pool statistic for each target; mean across targets.
$\mathrm{JSD}_{\mathrm{bb}}$, $\mathrm{JSD}_{\mathrm{sc}}$	Pool angles across all candidates and targets in a run, then take the unweighted mean of residue-type/angle profile distances within each angle family.

Ligand RMSD (L-RMSD) measures peptide Cα recovery after one Kabsch superposition of the peptide:

$$\text{L-RMSD} = \min_{\boldsymbol{Q}\in\mathrm{SO}(3),\,\boldsymbol{b}\in\mathbb{R}^3} \left[\frac{1}{n_\alpha}\sum_{i=1}^{n_\alpha} \|\boldsymbol{Q}\boldsymbol{x}'_i + \boldsymbol{b} - \boldsymbol{x}_i\|_2^2\right]^{1/2}. \tag{37}$$

Mapped-atom RMSD (M-RMSD). Mapped-atom RMSD extends peptide-aligned coordinate recovery to atoms paired by name within corresponding residues. When corresponding residues have the same amino-acid type, all atom names present in both peptides are paired; substitutions restrict correspondence to shared backbone atoms. One Kabsch superposition is applied to the resulting n_{map} pairs:

$$\text{M-RMSD} = \min_{\boldsymbol{Q}\in\mathrm{SO}(3),\,\boldsymbol{b}\in\mathbb{R}^3} \left[\frac{1}{n_{\mathrm{map}}}\sum_{a=1}^{n_{\mathrm{map}}} \|\boldsymbol{Q}\boldsymbol{x}'_a + \boldsymbol{b} - \boldsymbol{x}_a\|_2^2\right]^{1/2}. \tag{38}$$

Only paired atoms enter the sums and denominators above. C-RMSD, L-RMSD, and M-RMSD are reported in Å.

TM-score. TM-score (Zhang & Skolnick, 2004) evaluates peptide Cα structural similarity using the Zhang Lab implementation.

D.3 Contact and interface evaluation

Intrapeptide contact F1. Intrapeptide contact F1 measures recovery of peptide-internal contacts. Let Γ_{gen} and Γ_{ref} denote the generated and reference contact sets. A residue pair (i, j) is considered a contact when $|i - j| > 2$ and their Cα–Cα distance is at most 8 Å:

$$\mathrm{F1}_{\mathrm{intra}} = \frac{2|\Gamma_{\mathrm{gen}} \cap \Gamma_{\mathrm{ref}}|}{|\Gamma_{\mathrm{gen}}| + |\Gamma_{\mathrm{ref}}|}. \tag{39}$$

When both contact sets are empty, the score is defined as one.

DockQ. Target–peptide interface recovery is evaluated with DockQ v2.1.3 (Basu & Wallner, 2016).

Rosetta ΔG and improvement percentage (IMP). For each generated candidate and native complex, we compute `dG_separated` after each of three independent Rosetta relaxations and report their mean as Rosetta ΔG, in Rosetta energy units (REU) (Alford et al., 2017). Relaxation uses FastRelax with `ref2015`, allowing peptide-backbone motion and side-chain relaxation in the

peptide and its neighborhood. Thus, for candidate k of target j,

$$\Delta G_{jk} = \frac{1}{3}\sum_{r=1}^{3} \Delta G_{jk}^{(r)}, \qquad \Delta G_j^{\text{native}} = \frac{1}{3}\sum_{r=1}^{3} \Delta G_{j,\text{native}}^{(r)}. \tag{40}$$

Improvement percentage (IMP) reports the percentage of targets for which at least one candidate has a Rosetta ΔG no higher than that of the native complex:

$$\text{IMP}(\%) = \frac{100}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} \mathbb{I}\left[\min_{k\in\mathcal{K}_j} \Delta G_{jk} \le \Delta G_j^{\text{native}}\right], \tag{41}$$

where $\mathcal{J}$ indexes evaluated targets and $\mathcal{K}_j$ is the candidate pool for target j within one evaluation run.

D.4 Clash rates, dihedral distributions, and diversity

Cα clash rates. Following UniMoMo (Kong et al., 2025), we flag Cα pairs closer than 3.6574 Å. Clash_{in} is the fraction of peptide residues with at least one such peptide neighbor, excluding self-pairs and immediately adjacent residues along the sequence. $\text{Clash}_{\text{out}}$ is the fraction of peptide residues with at least one such neighbor in the target chains. Both rates use the number of peptide residues with Cα coordinates as the denominator and are reported as percentages.

Dihedral-distribution distances (JSD). Pooling angles from all generated peptides in a run gives one distribution for each amino-acid type and dihedral type: ϕ, ψ, and ω for the backbone, and available χ_1–χ_4 angles for side chains. Angles are discretized into 10° bins and compared with the fixed peptide reference profiles distributed with the UniMoMo evaluator (Kong et al., 2025). For generated and reference distributions $\boldsymbol{p}$ and $\boldsymbol{q}$, with $\boldsymbol{m} = (\boldsymbol{p}+\boldsymbol{q})/2$, the Jensen–Shannon distance is

$$d_{\text{JS}}(\boldsymbol{p},\boldsymbol{q}) = \left[\tfrac{1}{2}D_{\text{KL}}(\boldsymbol{p}\|\boldsymbol{m}) + \tfrac{1}{2}D_{\text{KL}}(\boldsymbol{q}\|\boldsymbol{m})\right]^{1/2}, \tag{42}$$

using natural logarithms. JSD_{bb} and JSD_{sc} are the unweighted means over backbone and side-chain residue-type/angle profiles, respectively, including only profiles with generated observations.

Sequence and pose diversity. Within each target's candidate pool, single-linkage clustering groups sequences at distance threshold 0.6 and peptide poses at RMSD threshold 2 Å. Sequence distance is one minus globally aligned identity, averaged over the two alignment directions using the AAR settings above. Pose distance is peptide Cα RMSD in the fixed target coordinate frame, without peptide superposition. For a pool of N candidates with K_{seq} sequence clusters and K_{pose} pose clusters,

$$D_{\text{seq}} = \frac{K_{\text{seq}}}{N}, \qquad D_{\text{pose}} = \frac{K_{\text{pose}}}{N}, \qquad D_{\text{codesign}} = \sqrt{D_{\text{seq}} D_{\text{pose}}}. \tag{43}$$

Co-design diversity is computed separately for each target before averaging across targets.

E Benchmark Implementations and Additional Results

E.1 Baseline implementations

Each baseline is retrained once on the common PepDataBench training split using its released implementation. We retain the released architectures and training objectives, with method-specific training and sampling settings summarized below. All methods use the same LNR targets, reference peptide lengths, and 100-candidate budget per target specified in Section 4.1. Generated candidates are evaluated using the common metrics and aggregation rules in Appendix D.

PepGLAD. PepGLAD (Kong et al., 2024) is retrained using its released two-stage geometric latent-diffusion configuration. The autoencoder and latent diffusion model are trained for at most 100 and 500 epochs, respectively. Sampling uses the 100-step diffusion process and the released latent-guidance settings for peptide co-design.

PepFlow. **PepFlow** (Li et al., 2024) is retrained using its released multimodal flow-matching formulation over sequence, backbone, and side-chain variables. We retain the default model and optimizer hyperparameters and use Euler integration with 200 equally spaced time points for sampling.

PepTri. **PepTri** (Nguyen et al., 2026) is retrained using its released peptide co-design configuration. The autoencoder and latent diffusion model are trained for at most 100 and 700 epochs, respectively, following the released training launcher used in our reproduction. We retain the 100-step diffusion process and the method's physics-, evolutionary-, and mutual-information-based guidance settings.

UniMoMo. **UniMoMo** (Kong et al., 2025) is retrained using its released iterative autoencoder and latent-diffusion architecture. We use the single-domain peptide setting and train on the common PEPDATABENCH split. The autoencoder and latent diffusion model are trained for at most 250 and 500 epochs, respectively. Sampling follows the released peptide configuration with a 100-step diffusion process.

PRIME. **PRIME** (Tian et al., 2026) is retrained using its released peptide autoencoder and latent-diffusion configuration with the frozen pretrained EPT encoder (Jiao et al., 2026b). The autoencoder and latent diffusion model are trained for at most 250 and 500 epochs, respectively. We retain the 100-step diffusion process and the method-specific exploratory sampling procedure, including the Graph-Laplacian and MC-CADS components.

E.2 RUN-LEVEL RESULTS AND SINGLE-RUN COMPARISON

The PepRAE entries in Tables 1 and 10 are means over three independently initialized diffusion runs sharing one selected Boltz-2 autoencoder, whereas each baseline is represented by one retrained model. Table 9 reports the individual PepRAE runs together with their mean and sample SD. The SDs summarize variation across diffusion training runs conditional on the fixed autoencoder. Run identifiers and initialization seeds follow Table 20.

As a supplementary sensitivity analysis, we compare diffusion Run 1 (seed 12) with the same baseline entries in Table 1. This removes averaging over PepRAE diffusion runs while leaving the evaluation targets, candidate budget, and metric aggregation unchanged.

In this single-run comparison, PepRAE retains the lowest C-RMSD, L-RMSD, and M-RMSD, the lowest Rosetta ΔG, and the highest IMP. $F1_{\mathrm{intra}}$ remains second to UniMoMo. AAR ranks second, at 39.18% compared with PepTri's 39.23%, while DockQ ranks third behind UniMoMo and PRIME. Thus, the main geometric-recovery and Rosetta-score findings persist without averaging across PepRAE diffusion runs, although the AAR and DockQ rankings change. Because each baseline is trained once, this analysis addresses the effect of diffusion-run averaging on the PepRAE side but does not estimate baseline training variability.

E.3 ADDITIONAL BENCHMARK METRICS

Table 10 complements the primary benchmark with additional measures of structural recovery, Cα clashes, dihedral distributions, and candidate-pool diversity. PepRAE achieves the highest TM-score (0.530) and the lowest intrapeptide and target–peptide Cα clash rates (0.46% and 1.51%, respectively). The clash rates are averaged over all candidates within each target before averaging across targets, characterizing the full candidate pools before relaxation.

For dihedral distributions, PepRAE achieves the lowest side-chain distance ($\mathrm{JSD}_{\mathrm{sc}} = 0.302$), closely followed by PRIME (0.304). Its backbone distance ($\mathrm{JSD}_{\mathrm{bb}} = 0.344$) is second to PepFlow (0.260). These results complement the coordinate-recovery metrics with comparisons of local backbone and side-chain conformations against the evaluator's fixed reference distributions.

Co-design diversity is 0.572, consistent with the diversity profile in Table 1: sequence diversity remains competitive, whereas pose diversity is lower than that of most baselines.

Table 9: **Run-level results for the primary PepRAE benchmark.** The same selected Boltz-2 autoencoder is fixed across the three diffusion runs. Mean and sample SD summarize the run-level metrics. RMSDs are in Å, ΔG is in REU, and AAR, IMP, and clash rates are percentages; their SDs are in percentage points.

(a) Core recovery, interface, and Rosetta metrics

Run	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	ΔG ↓	IMP↑
1	39.18%	2.395	1.155	1.408	0.719	-30.431	32.61%
2	40.67%	2.560	1.205	1.542	0.721	-29.942	25.00%
3	39.33%	2.340	1.280	1.473	0.747	-30.110	30.43%
Mean	**39.73%**	**2.432**	**1.213**	**1.474**	**0.729**	**-30.161**	**29.35%**
SD	0.82%	0.114	0.063	0.067	0.016	0.249	3.92%

(b) Structural quality

Run	TM-score↑	$F1_{\text{intra}}$ ↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓
1	0.528	0.884	0.41%	1.65%
2	0.531	0.901	0.41%	1.44%
3	0.532	0.900	0.55%	1.42%
Mean	**0.530**	**0.895**	**0.46%**	**1.51%**
SD	0.002	0.009	0.08%	0.13%

(c) Distribution and diversity

Run	JSD_{bb} ↓	JSD_{sc} ↓	D_{seq} ↑	D_{pose} ↑	D_{codesign} ↑
1	0.342	0.299	0.635	0.536	0.553
2	0.344	0.303	0.643	0.573	0.582
3	0.346	0.304	0.649	0.571	0.581
Mean	**0.344**	**0.302**	**0.642**	**0.560**	**0.572**
SD	0.002	0.002	0.007	0.021	0.017

Table 10: **Additional benchmark metrics on LNR.** These metrics complement Table 1. Boldface and underlining denote the best and second-best reported point estimates, respectively. Clash rates are percentages.

Method	TM-score↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓	JSD_{bb} ↓	JSD_{sc} ↓	D_{codesign} ↑
PepGLAD	0.352	2.93%	1.93%	0.528	0.403	<u>0.643</u>
PepFlow	0.458	3.55%	5.15%	**0.260**	0.735	0.564
PepTri	0.376	12.58%	6.54%	0.534	0.395	**0.711**
UniMoMo	<u>0.514</u>	<u>1.61%</u>	<u>1.92%</u>	0.350	0.327	0.619
PRIME	0.509	2.14%	6.25%	0.355	<u>0.304</u>	0.573
PepRAE	**0.530**	**0.46%**	**1.51%**	<u>0.344</u>	**0.302**	0.572

E.4 Performance by peptide length

Figure 4 compares PepRAE with UniMoMo and PRIME, the two baselines with the lowest overall C-RMSD in Table 1. AAR and DockQ decrease across the four length ranges for all three methods, while C-RMSD increases through the 11–15-residue range. In the 16–25-residue range, PepRAE has a similar C-RMSD to its preceding range and a lower median C-RMSD than the two baselines, whose errors continue to increase. Larger sequence and conformational search spaces may contribute to these trends under the fixed 100-candidate budget. The ranges contain different targets and unequal sample sizes, so this comparison describes length-associated variation rather than isolating the effect of peptide length.

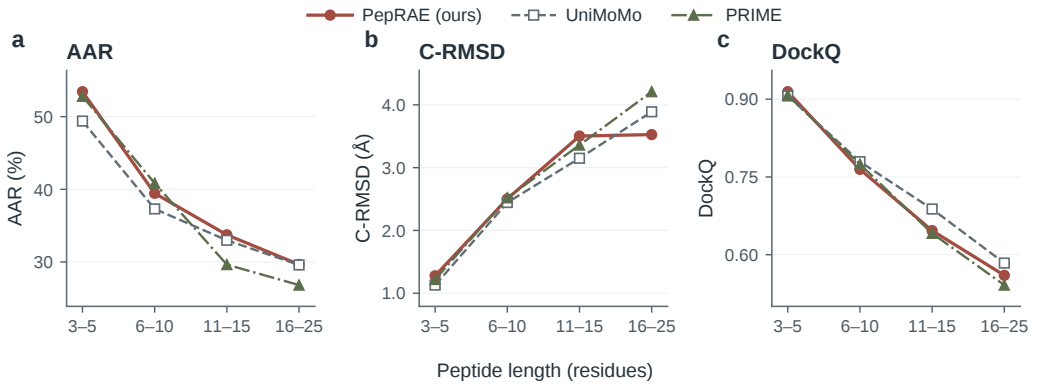


Figure 4: **Generation performance across peptide-length ranges.** The ranges 3–5, 6–10, 11–15, and 16–25 contain 22, 28, 25, and 17 LNR targets, respectively. The final range combines 16–20 and 21–25 residues because the latter contains only two targets. Within each range, target-level AAR and DockQ are averaged, whereas C-RMSD is summarized by the median.

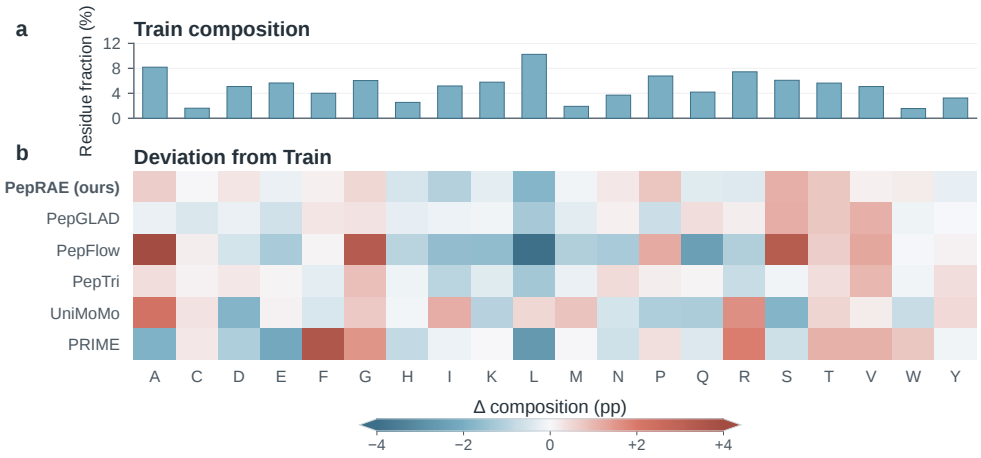


Figure 5: **Amino-acid composition of benchmark candidate pools.** (a) Residue-weighted amino-acid composition of peptides in the PEPDATABENCH training split. (b) Percentage-point deviation of each method's generated peptide composition from the training-peptide reference across the 20 canonical amino acids. Positive values indicate enrichment and negative values indicate depletion relative to this reference. The color scale saturates at ± 4 percentage points.

E.5 AMINO-ACID COMPOSITION

We compare the pooled amino-acid composition of the generated peptide candidate pools with that of peptides in the PEPDATABENCH training split, using the 20 canonical amino acids. Figure 5 shows the training-peptide residue frequencies and method-specific deviations from this reference. The comparison summarizes residue usage across the candidate pools.

F MATCHED REPRESENTATION-SOURCE GENERATION RESULTS

This appendix reports the ESM-2 and LightEnc diffusion runs in the matched-source comparison of Section 4.5. The Boltz-2 configuration reuses the three primary benchmark runs in Table 9.

Table 11: **Run-level core generation metrics for additional representation sources.** Mean and sample SD summarize the three diffusion runs for each source. RMSDs are in Å, ΔG is in REU, and AAR and IMP are percentages; their SDs are in percentage points.

Source	Run	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	ΔG ↓	IMP↑
ESM-2	1	40.75%	2.505	1.295	1.490	0.708	-29.683	25.00%
	2	40.39%	2.715	1.270	1.488	0.740	-30.347	29.35%
	3	40.11%	2.365	1.285	1.519	0.733	-30.377	30.43%
	Mean	**40.41%**	**2.528**	**1.283**	**1.499**	**0.727**	**-30.136**	**28.26%**
	SD	0.32%	0.176	0.013	0.017	0.017	0.392	2.88%
LightEnc	1	37.68%	2.445	1.085	1.321	0.743	-30.889	28.26%
	2	38.42%	2.450	1.190	1.450	0.737	-30.038	30.43%
	3	38.36%	2.560	1.285	1.451	0.759	-29.881	27.17%
	Mean	**38.15%**	**2.485**	**1.187**	**1.408**	**0.746**	**-30.269**	**28.62%**
	SD	0.41%	0.065	0.100	0.075	0.011	0.542	1.66%

F.1 Run-level generation results

Tables 11 and 12 report the six ESM-2 and LightEnc diffusion runs underlying the aggregate cross-source generation results in Table 4. Together with the Boltz-2 runs in Table 9, these constitute the nine diffusion runs used in the matched-source comparison.

Appendix C specifies source extraction and preprocessing. Appendices J.1 and J.2 give the shared downstream architecture, training protocol, and run identifiers.

Table 12: **Supplementary run-level generation metrics for additional representation sources.** The same six ESM-2 and LightEnc diffusion runs as in Table 11 are reported. Panel (a) summarizes structural quality, while panel (b) reports dihedral-distribution agreement and candidate-pool diversity. Mean and sample SD summarize the three runs for each source. Clash rates are percentages and their SDs are in percentage points.

(a) Structural quality

Source	Run	TM-score↑	$F1_{\text{intra}}$ ↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓
ESM-2	1	0.542	0.892	0.48%	1.08%
	2	0.542	0.901	0.37%	1.17%
	3	0.532	0.879	0.56%	1.11%
	Mean	**0.539**	**0.891**	**0.47%**	**1.12%**
	SD	0.006	0.011	0.10%	0.04%
LightEnc	1	0.552	0.901	0.31%	0.74%
	2	0.538	0.899	0.29%	0.94%
	3	0.520	0.908	0.72%	1.08%
	Mean	**0.536**	**0.903**	**0.44%**	**0.92%**
	SD	0.016	0.005	0.24%	0.17%

(b) Distribution and diversity

Source	Run	JSD_{bb} ↓	JSD_{sc} ↓	D_{seq} ↑	D_{pose} ↑	D_{codesign} ↑
ESM-2	1	0.333	0.323	0.626	0.503	0.532
	2	0.338	0.326	0.640	0.527	0.547
	3	0.336	0.322	0.645	0.497	0.529
	Mean	**0.336**	**0.324**	**0.637**	**0.509**	**0.536**
	SD	0.003	0.002	0.010	0.015	0.009
LightEnc	1	0.329	0.310	0.636	0.519	0.544
	2	0.332	0.312	0.623	0.521	0.534
	3	0.339	0.315	0.636	0.577	0.577
	Mean	**0.333**	**0.312**	**0.632**	**0.539**	**0.552**
	SD	0.005	0.003	0.008	0.033	0.023

G Ablation Details and Variation

This appendix records the precise interventions used in the core ablations of Section 4.4 and the corresponding reconstruction results and run-to-run variation in generation. Reconstruction metrics follow Appendix H, diffusion run identifiers follow Table 20, and generation metrics use the aggregation rules in Appendix D. The main-text ablation tables report three-run generation means; the generation tables below pair those means with sample standard deviations. These SDs measure diffusion-initialization variation conditional on each selected autoencoder, rather than variation across autoencoder training runs. RMSDs are in Å, ΔG is in REU, and AAR and IMP are percentages; their SDs are in percentage points.

G.1 Representation-adaptation controls

All representation-adaptation controls use the Boltz-2 source and the same downstream architecture, feature-latent width, spatial pathway, training data, and generation protocol as the corresponding PepRAE configuration. Each control retrains the autoencoder under the specified intervention; the validation-selected autoencoder for that configuration is then held fixed across three independently trained diffusion models.

The *No* $\mathcal{L}_{\text{rep}}$ control removes the representation-reconstruction term while retaining peptide reconstruction and KL regularization.

Table 13: **Reconstruction under representation-adaptation ablations.** Deterministic posterior-mean decoding on 1,064 validation complexes, using one selected Boltz-2 autoencoder per configuration. Checks identify losses that update representation-decoder parameters; the $\mathcal{L}_{\text{rep}}$-only control still propagates peptide-loss gradients to the feature latents and encoder. Aligned RMSDs use peptide-only Kabsch superposition and are in Å; atom RMSD uses corresponding non-hydrogen atoms. Boldface and underlining mark the best and second-best values.

Configuration	$\mathcal{L}_{\text{rep}} \to \mathcal{D}_\xi^{\text{rep}}$	$\mathcal{L}_{\text{mol}} \to \mathcal{D}_\xi^{\text{rep}}$	Repr. Smooth-L1↓	Repr. cosine↑	Residue acc.↑	Aligned Cα RMSD↓	Aligned atom RMSD↓
PepRAE	✓	✓	**0.1937**	**0.7792**	**99.89%**	**0.4740**	**1.7886**
No $\mathcal{L}_{\text{rep}}$	×	✓	0.5162	-0.0023	99.81%	0.5021	1.9554
$\mathcal{L}_{\text{rep}}$-only $\mathcal{D}_\xi^{\text{rep}}$	✓	×	<u>0.2136</u>	<u>0.7638</u>	<u>99.84%</u>	<u>0.4825</u>	<u>1.8036</u>

The $\mathcal{L}_{\text{rep}}$-only $\mathcal{D}_\xi^{\text{rep}}$ control prevents peptide-reconstruction loss from updating the representation-decoder parameters. Backpropagation through $\mathcal{D}_\xi^{\text{rep}}$ to the feature-latent inputs $\boldsymbol{Z}_{h,C}$ and encoder is retained. The representation decoder remains trained by $\mathcal{L}_{\text{rep}}$. Both controls retain the forward pathway through $\mathcal{D}_\xi^{\text{rep}}$.

Reconstruction diagnostics. Table 13 reports deterministic reconstruction for the selected autoencoder in each configuration, averaged across the 1,064 validation complexes. Without $\mathcal{L}_{\text{rep}}$, residue accuracy remains nearly unchanged (99.81% versus 99.89%), whereas peptide-row source-feature cosine similarity falls from 0.7792 to -0.0023 and both peptide-aligned RMSDs increase. Thus, accurate residue reconstruction alone does not ensure fidelity at the reconstructed source-feature interface. Restricting representation-decoder updates to $\mathcal{L}_{\text{rep}}$ produces smaller degradations in these reconstruction metrics.

Generation variation. Table 14 gives the means and sample SDs corresponding to the generation results in Table 2. The reconstruction and generation results support the two training choices under the evaluated setup; they do not isolate which retained source-feature information accounts for the generation differences.

Table 14: **Generation means and variation for representation-adaptation ablations.** Mean and sample SD are computed across three diffusion runs for each configuration.

Statistic	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	ΔG ↓	IMP↑
PepRAE							
Mean	**39.73%**	**2.432**	**1.213**	**1.474**	**0.729**	**-30.161**	**29.35%**
SD	0.82%	0.114	0.063	0.067	0.016	0.249	3.92%
No $\mathcal{L}_{\text{rep}}$							
Mean	**38.91%**	**2.612**	**1.343**	**1.493**	**0.724**	**-27.946**	**25.00%**
SD	0.42%	0.188	0.053	0.038	0.013	0.615	1.26%
$\mathcal{L}_{\text{rep}}$-only $\mathcal{D}_\xi^{\text{rep}}$							
Mean	**38.63%**	**2.682**	**1.330**	**1.523**	**0.717**	**-28.692**	**26.81%**
SD	0.56%	0.139	0.060	0.046	0.008	0.626	1.26%

G.2 TARGET-ADAPTER CONTROLS

All target-adapter controls use the same fixed validation-selected Boltz-2 autoencoder; only the training signal received by the target adapter is changed. In PepRAE, the adapter is trained by explicit alignment supervision while denoising gradients are stopped at the adapter output. The *No supervision* control removes alignment while retaining this gradient stop, leaving the adapter without task-related training signal. The *Denoising-supervised* control also removes alignment but

Table 15: **Training signals received by the target adapter.** Checks indicate active supervision or gradient paths. All configurations share the same fixed Boltz-2 autoencoder.

Configuration	Alignment supervision	Denoising gradient
PepRAE	✓	×
No supervision	×	×
Denoising-supervised	×	✓

Table 16: **Generation means and variation for target-adapter ablations.** Mean and sample SD are computed across three diffusion runs for each configuration.

Statistic	AAR↑	C-RMSD↓	L-RMSD↓	M-RMSD↓	DockQ↑	ΔG ↓	IMP↑
PepRAE							
Mean	**39.73%**	**2.432**	**1.213**	**1.474**	**0.729**	**-30.161**	**29.35%**
SD	0.82%	0.114	0.063	0.067	0.016	0.249	3.92%
No supervision							
Mean	**37.74%**	**2.767**	**1.343**	**1.586**	**0.716**	**-26.215**	**23.55%**
SD	0.54%	0.239	0.037	0.065	0.014	1.015	2.51%
Denoising-supervised							
Mean	**38.02%**	**2.780**	**1.382**	**1.547**	**0.701**	**-27.583**	**25.72%**
SD	0.66%	0.048	0.091	0.021	0.015	0.235	2.26%

removes the gradient stop so that the adapter is updated through the denoising objective. All other target-conditioning pathways and generation settings are unchanged across these controls.

Table 16 reports the generation means and sample standard deviations corresponding to Table 3.

H Reconstruction Evaluation and Autoencoder Robustness

H.1 Evaluation protocol and metrics

Reconstruction is evaluated on the 1,064 validation complexes using deterministic posterior means for both feature and spatial latents, without posterior sampling. Decoding follows the standard peptide-decoding procedure described in Appendix A.2. All metrics are computed separately for each complex and then averaged equally across the validation set.

Source-feature reconstruction. Although $\mathcal{L}_{\text{rep}}$ supervises all retained complex rows during autoencoder training, the source-feature reconstruction diagnostics reported here are evaluated on peptide rows. For each peptide residue, the reference source feature is normalized across its d_m channels using LayerNorm without learned affine parameters. We report Smooth-L1 loss and residue-wise cosine similarity, averaging first over peptide rows within each complex and then equally across the 1,064 validation complexes. Because these quantities are defined in source-specific feature spaces, they are treated as reconstruction diagnostics within each source rather than as direct measures of representation quality across sources.

Peptide reconstruction. Residue accuracy is the fraction of peptide positions whose decoded top-1 residue identity matches the observed identity at the corresponding position. For peptide-aligned Cα RMSD, predicted and observed Cα atoms are paired by residue position and compared after a peptide-only Kabsch superposition. Peptide-aligned atom RMSD is computed after one peptide-only Kabsch superposition over non-hydrogen atoms matched by atom name within corresponding residue positions.

The aligned Cα reconstruction metric uses the same peptide-only alignment notion as generation L-RMSD, but the reported statistics are not directly comparable. Reconstruction evaluates one prediction per validation complex and reports the arithmetic mean across 1,064 complexes, whereas the LNR

benchmark evaluates 100 candidates per target before metric-specific candidate-pool aggregation over 92 targets. The aligned atom reconstruction metric is likewise peptide-aligned but is not identical to generation M-RMSD, which uses the mapped-atom correspondence defined by the generation evaluator in Appendix D.

Matched heavy-atom coverage is defined as the fraction of native peptide non-hydrogen atoms with a corresponding decoded atom under the matching rule above. Across the selected Boltz-2, ESM-2, and LightEnc autoencoders, residue accuracy is 99.89–100.00% and matched heavy-atom coverage is 99.61–99.63%. Thus, the reported atom RMSD evaluates nearly the complete peptide heavy-atom structure in these reconstructions.

H.2 Reconstruction across representation sources

Table 17 reports the complete reconstruction results for the selected source-specific autoencoders in Section 4.5, using the protocol above. All three yield residue accuracy of 99.89–100.00% and peptide-aligned Cα RMSDs of 0.471–0.492 Å. Their source-feature reconstruction scores are interpreted within each source because the reconstruction targets differ.

Table 17: **Reconstruction across representation sources.** Deterministic posterior-mean decoding on 1,064 validation complexes, using each source's selected autoencoder. Aligned RMSDs use peptide-only Kabsch superposition and are in Å; atom RMSD uses corresponding non-hydrogen atoms. Source-feature Smooth-L1 and cosine values compare reconstructions with each source's own features and are not rankings of source quality.

Source	Repr. Smooth-L1↓	Repr. cosine↑	Residue acc.↑	Aligned Cα RMSD↓	Aligned atom RMSD↓
Boltz-2	0.1937	0.7792	99.89%	0.4740	1.7886
ESM-2	0.1162	0.8684	100.00%	0.4916	1.8279
LightEnc	0.0471	0.9513	100.00%	0.4710	1.8042

H.3 Autoencoder initialization robustness

We assess reconstruction robustness across three independently initialized Boltz-2 autoencoders using the same evaluation protocol. Run identifiers follow Table 20; Run 1 is the validation-selected autoencoder used for the primary Boltz-2 experiments.

Table 18: **Autoencoder reconstruction across Boltz-2 initializations.** Run 1 is the autoencoder used for the primary Boltz-2 experiments. Mean and sample SD summarize the three autoencoder runs. RMSDs are in Å; residue accuracy is a percentage and its SD is in percentage points.

Run	Repr. Smooth-L1↓	Repr. cosine↑	Residue acc.↑	Aligned Cα RMSD↓	Aligned atom RMSD↓
1	0.1937	0.7792	99.89%	0.4740	1.7886
2	0.1900	0.7804	99.80%	0.4718	1.7340
3	0.1924	0.7779	99.83%	0.4702	1.7688
Mean	**0.1920**	**0.7792**	**99.84%**	**0.4720**	**1.7638**
SD	0.0019	0.0012	0.05%	0.0019	0.0276

I Latent-space Analysis

This appendix describes the analyses in Section 4.3 and Figure 2, using the same fixed, validation-selected Boltz-2 autoencoder.

I.1 FEATURE-LATENT ORGANIZATION

Figure 2a visualizes deterministic peptide feature posterior means from the validation set using t-SNE, with residues colored by amino-acid identity. Neighborhood purity and linear classification are evaluated in the original eight-dimensional feature-latent space. Cross-complex kNN@20 purity measures amino-acid identity agreement among 20 nearest neighbors while excluding residues from the query's own complex, yielding 0.9108. A linear probe achieves 99.66% identity classification accuracy on held-out targets. These original-space measurements assess identity organization and linear readability; the t-SNE projection is used only as a qualitative visualization of the latent organization.

I.2 LATENT PERTURBATION PROTOCOL

Paired decoding. Each perturbed output is compared with unperturbed decoding from the same autoencoder and the same input complex. The unperturbed peptide inputs use deterministic feature posterior means $\boldsymbol{\mu}_{h,C}[P]$ and spatial posterior means $\boldsymbol{X}_P^{\alpha}$. Target-side inputs, unaffected latent inputs, and decoder randomness are held fixed. Identity changes refer to differences between the two decoded top-1 residue identities. All coordinate responses are measured in the fixed target frame, without peptide superposition or re-centering. These measurements quantify responses to perturbations, rather than reconstruction errors relative to the native peptide.

Single-residue perturbations (panels b,c). For each of 32 validation targets, we perturb the row of the deterministic peptide feature posterior means $\boldsymbol{\mu}_{h,C}[P]$ corresponding to the lower-middle peptide residue, with zero-based index $i_* = \lfloor (L-1)/2 \rfloor$ for peptide length L. Each target is evaluated with four fixed QR-orthogonal, unit-RMS directions, scaled channel-wise by the standard deviations of deterministic peptide feature posterior means from the 9,113 training complexes. For direction $\boldsymbol{d}^{(k)}$ and channel-wise training SD vector $\boldsymbol{s}_h$, we add $a(\boldsymbol{s}_h \odot \boldsymbol{d}^{(k)})$ to the selected row, with $a \in \{0.1, 0.25, 0.5, 1, 2\}$. All other feature rows and all spatial inputs remain fixed. Panel b reports the identity-change rate at the perturbed residue and the fraction of changed identities among the remaining peptide residues. Panel c reports the mean Euclidean displacement of backbone N, Cα, C, and O atoms, separately for the same two residue groups. For each metric, we average over directions within each target and then equally over targets. At $a = 2$, the identity-change rate is 0.789 at the perturbed residue, with no observed top-1 identity changes elsewhere. Backbone displacements nevertheless extend to other residues, so localized top-1 changes do not imply that other residues are unaffected.

All-residue perturbations (panels d,e). For 64 validation targets, we perturb all peptide rows of one field while holding the other field fixed. For the feature field, independent Gaussian noise with SD λ in raw feature-latent units is added to each row and channel of the deterministic posterior means $\boldsymbol{\mu}_{h,C}[P]$. For the spatial field, perturbations are applied to the normalized deterministic spatial posterior means, $\boldsymbol{U}_{x,P}^{(0)} = \Phi_T(\boldsymbol{X}_P^{\alpha})$. Independent Gaussian noise with SD $\lambda/10$ is added to each peptide row and Cartesian component. The perturbed coordinates are transformed by Φ_T^{-1} before decoding, corresponding to an input-noise SD of λ Å per Cartesian component under $\kappa_x = 10$ Å. No reverse-diffusion steps are performed in this diagnostic. The two fields are not norm-matched; equal values of λ do not represent equal perturbation norms across fields, so the comparison is used to characterize response types rather than absolute perturbation sensitivity. For each target and field, the same noise tensor is reused across magnitudes. Panel d reports the peptide-wide fraction of top-1 identities changed from clean decoding, averaged equally over targets. Panel e reports peptide Cα RMSD to clean decoding, computed separately for each target and then averaged equally over targets. Spatial noise is not centered across peptide rows, so coordinate responses can include changes in overall placement as well as internal geometry.

Uncertainty. Curves show means across targets. Shading denotes pointwise 95% percentile intervals from 10,000 PDB-accession-cluster bootstrap resamples. These intervals describe variability across PDB-accession clusters in the evaluated set for the fixed autoencoder, rather than variability across autoencoder training runs.

I.3 Case Study

Figure 6 visualizes two representative target-only generation examples on 3T6J and 6FQ4. For each target, the reference peptide is shown in purple and the PepRAE-generated peptide in salmon under the same target structure and viewing angle. The two cases complement the aggregate evaluation by illustrating the generated binding conformations for a compact peptide and a longer 16-residue peptide.

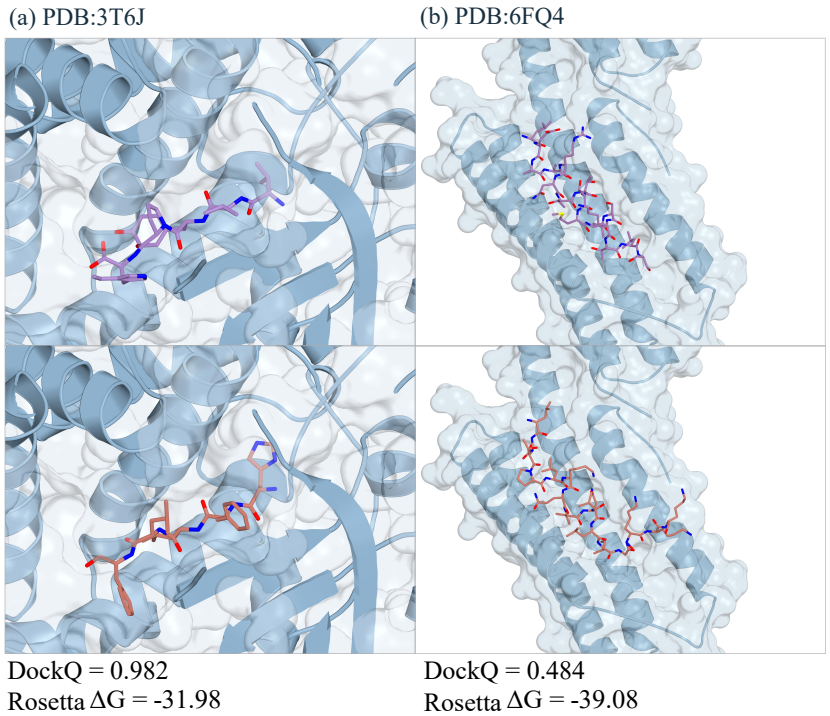


Figure 6: **Representative target-only generation examples.** (a) 3T6J and (b) 6FQ4. Reference peptides are shown in purple (top), while PepRAE-generated peptides are shown in salmon (bottom), using the same target structure and viewing angle within each case. DockQ and the Rosetta interface ΔG of the generated peptide are reported below each target.

J Implementation and Reproducibility Details

J.1 Model architecture

The matched Boltz-2, ESM-2, and LightEnc configurations use source features of width $d_m = 384$ after fixed preprocessing and feature latents of width $d_h = 8$. A separate PepRAE model is trained for each representation source, with the downstream architecture sizes and training objectives held fixed across the three configurations. Table 19 summarizes the main downstream modules. Source extraction, preprocessing, and LightEnc pretraining are described in Appendix C.

The target adapter $\mathcal{A}_\eta$ applies LayerNorm followed by a linear projection from d_m to d_h. Its outputs are the fixed target feature conditions $\boldsymbol{U}_{h,T}$ used during diffusion training and generation. Peptide decoding and diffusion denoising use EPT-based geometric modules with residue-block representations (Jiao et al., 2026b; Kong et al., 2025).

J.2 Optimization and checkpoint selection

For each matched-source configuration used in the main experiments, the source-specific autoencoder is trained using Run 1, with its checkpoint selected by validation performance. For the Boltz-2

Table 19: **Downstream architecture sizes for the matched-source configurations.** Widths refer to hidden dimensions within each module.

Module	Layers	Width	Heads
Encoder $\mathcal{E}_\phi$ (Transformer)	1	384	8
Representation decoder $\mathcal{D}_\xi^{\text{rep}}$ (Transformer)	6	384	8
Peptide decoder $\mathcal{D}_\psi^{\text{mol}}$ (EPT)	6	512	8
Diffusion denoiser (EPT)	6	512	8

Table 20: **Training-run identifiers and initialization seeds.** Run identifiers are defined consistently across training stages. Main-experiment autoencoders use Run 1; the autoencoder robustness analysis additionally evaluates Runs 2 and 3. Diffusion experiments use Runs 1–3.

Run	Initialization seed
1	12
2	42
3	2027

autoencoder initialization-robustness analysis in Appendix H.3, two additional autoencoders are trained as Runs 2 and 3 using the same autoencoder training settings.

For the primary PepRAE benchmark, matched-source configurations, and main retrained ablations, one validation-selected autoencoder checkpoint is held fixed within each configuration while three diffusion models are trained independently as Runs 1–3. The source model and autoencoder remain frozen during diffusion training. Table 20 maps the run identifiers used at both training stages to their initialization seeds.

The primary diffusion run (Run 1) uses initialization seed 12 and is used for the single-run comparison in Appendix E.2; the main benchmark reports means across all three diffusion runs.

Each trained diffusion model is evaluated separately on LNR using the same generation seed (12), producing 100 candidates per target. This seed controls sampling rather than training initialization. Metrics are computed separately for each diffusion run; candidate pools are not pooled across runs. Run-level aggregation and variation are defined in Appendix D.

Both training stages use AdamW (Loshchilov & Hutter, 2019). Dynamic batch packing yields approximately 49 global optimizer updates per epoch. At the epoch limits, this corresponds to approximately 12,250 and 14,700 global optimizer updates for autoencoder and diffusion training, respectively. Table 21 summarizes the optimization settings for the matched-source configurations.

Each autoencoder and diffusion training run uses eight NVIDIA A800 GPUs in FP32. For a per-GPU minibatch $\mathcal{B}$, dynamic packing uses the cost

$$|\mathcal{B}| \left(\max_{i \in \mathcal{B}} n_i \right)^2$$

with a per-GPU budget of 1.2×10^7, where n_i is the number of atom nodes in the graph over the retained target residues and peptide for complex i.

Autoencoder checkpoints are selected using validation $\mathcal{L}_{\text{AE}}$. Diffusion checkpoints are selected using non-EMA validation $\mathcal{L}_{\text{diff}}$, and generation uses the corresponding exponential moving average (EMA) parameters.

Autoencoder loss weights and decoding settings are specified in Appendix A.2. Target-adapter alignment, gradient routing, and diffusion sampling are detailed in Appendix A.3. Metric aggregation and uncertainty reporting are described in Appendix D.

Table 21: **Optimization settings for the matched-source configurations.** Settings are shared across runs within each training stage; EMA denotes exponential moving average.

Setting	Autoencoder	Diffusion model
Maximum epochs	250	300
Approx. global optimizer updates / epoch	49	49
Approx. global optimizer updates at the epoch limit	$\approx 12{,}250$	$\approx 14{,}700$
Main-experiment run identifier(s)	Run 1	Runs 1–3
Optimizer	AdamW	AdamW
Initial learning rate	10^{-4}	10^{-4}
Learning-rate schedule	Constant	ReduceLROnPlateau
Weight decay	0.01	0.01
Validation frequency	Every 10 epochs	Every epoch
Gradient clipping	Global norm 1.0	None
KL warm-up	4,000 updates	—
EMA decay	—	0.996
EMA initialization	—	After 5,000 updates

K Additional Related Work

Additional peptide sequence–structure generators. Additional peptide generators explore alternative structural representations and generative formulations. PepBFN jointly models residue types and structural variables using Bayesian flow networks (Qian et al., 2026). PepHAR samples hot-spot residues and extends peptide fragments autoregressively, while SurfFlow jointly models peptide sequences, structures, and molecular surfaces (Li et al., 2025a; Wu et al., 2025). MEET develops a memory-efficient equivariant Transformer for scalable full-atom peptide generation and incorporates it into a variational autoencoding and latent-diffusion framework (Jiao et al., 2026a).

Pretrained structural representations for binder generation. PRIME derives molecular representations from atom types and coordinates using a frozen pretrained EPT encoder. These representations support its molecular-reconstruction and latent-diffusion framework (Tian et al., 2026; Jiao et al., 2026b).

PRIME and PepRAE therefore both employ feature and spatial latent variables within a two-stage latent generative framework, but differ in how the feature latents are obtained and adapted for molecular decoding. In PepRAE, feature latents are learned from residue-aligned protein-model representations extracted from sequence contexts through a fixed source interface, while peptide geometry is represented separately by explicit Cα-based spatial latents. A representation decoder maps the feature latents back to the source-feature space before peptide decoding. Consequently, representation reconstruction and peptide reconstruction jointly supervise the same feature-latent pathway, and the peptide decoder combines the reconstructed source features with the spatial latents and supplied target graph. After reconstruction training, target-conditioned diffusion jointly models the peptide feature and spatial latents with the autoencoder fixed.

Peptide sequence proposals and optimization. Other approaches focus on proposing, ranking, or optimizing peptide sequences for a target. PepMLM uses masked language modeling conditioned on the target sequence, while PepPrCLIP combines peptide-embedding perturbations with contrastive ranking (Chen et al., 2026; Bhat et al., 2025). High-PepBinder integrates protein language models with target-conditioned diffusion and an affinity classifier; PepCCD conditions a pretrained diffusion model on contrastively learned target representations (Mao et al., 2026; Zhang et al., 2026). BOND-PEP uses retrieved target-relevant peptide examples to condition autoregressive sequence generation

(Ding, 2026). EvoBind2 uses predicted complexes to guide sequence search for linear and cyclic peptides (Li et al., 2025b).

Broader protein and binder generation. Related geometric and latent generative models also extend beyond peptide-specific design. La-Proteina jointly generates explicit Cα coordinates and latents learned from sequences and atomic structures (Geffner et al., 2026). Proteina-Complexa extends this formulation to target-conditioned binder design with inference-time optimization (Didi et al., 2026). AnewOmni uses graph prompts to specify chemical, topological, and geometric constraints across molecular modalities (Kong et al., 2026). BoltzGen supports all-atom generation across binder and target types (Stark et al., 2025).

Prediction-guided binder design. Structure-prediction networks can also be adapted into generators or provide feedback for sequence optimization. RFdiffusion fine-tunes RoseTTAFold for protein backbone generation, with ProteinMPNN assigning sequences to the generated backbones (Watson et al., 2023; Dauparas et al., 2022). BindCraft optimizes binder sequences by backpropagating through AlphaFold2, followed by sequence redesign and filtering (Pacesa et al., 2025).

Generative adaptation across representation sources. Image-generation studies also examine reconstruction and generation with different representation encoders. RAE compares reconstruction and generation with DINOv2, SigLIP2, and MAE as pretrained representation encoders (Zheng et al., 2026). FAE evaluates DINOv2 and SigLIP2 within the same feature-autoencoding design for text-to-image generation (Gao et al., 2026). RepTok represents each image with a single continuous latent token and reports reconstruction and generation results with DINOv2, MAE, and CLIP backbones (Gui et al., 2026). Our cross-source analysis examines whether the same PepRAE formulation can be instantiated with distinct protein representation sources by training separate Boltz-2, ESM-2, and LightEnc variants with the same downstream architecture and training objectives.