PEPMAP: MULTIMODAL FLOW MAPS FOR FEW-STEP ALL-ATOM PEPTIDE BINDER DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing peptide binders for a target protein requires generating amino-acid sequences and bound conformations that are compatible with its binding site. Recent diffusion and flow-matching methods can jointly generate peptide sequences and structures, but their iterative sampling makes candidate generation costly. Few-step sampling is challenging because residue identities and backbone geometry must remain coordinated across large time intervals. We introduce **PepMap**, a pocket-conditioned multimodal Flow Map for few-step peptide sequence–structure co-design. PepMap learns finite-time transitions that jointly update continuous sequence states, residue positions, and backbone orientations. Flow-matching supervision guides local predictions, while multimodal semigroup consistency encourages agreement between direct and composed transitions. Geometric objectives regularize backbone geometry and peptide–pocket compatibility, and a conditional side-chain packing module produces full-atom peptide structures. On the LNR benchmark, PepMap improves structural recovery over iterative baselines at their standard sampling budgets while requiring substantially fewer network evaluations. These results support finite-time transport of coupled sequence and backbone states for efficient pocket-conditioned peptide design. Code is available at `https://anonymous.4open.science/r/PepMap-anonymous-E018/`.

1 INTRODUCTION

Peptide sequence plasticity and conformational diversity make peptides attractive for targeting protein–protein interaction interfaces (Craik et al., 2013; Muttenthaler et al., 2021). Designing a binder for a given receptor requires jointly determining its amino-acid sequence and bound conformation under pocket constraints. The sequence specifies residue chemistry, while the backbone and side chains determine how these residues are organized at the interface. Pocket-conditioned peptide design therefore requires modeling coupled sequence–structure distributions compatible with the receptor environment (Li et al., 2024).

Traditional computational peptide design relies on structural templates, conformational search, and energy optimization (Bhardwaj et al., 2016; Hosseinzadeh et al., 2021; Alford et al., 2017; Cao et al., 2022). Recent generative models (Watson et al., 2023; Rettie et al., 2025) learn receptor-conditioned peptide distributions from structural data. Diffusion-based methods include DiffPepBuilder (Wang et al., 2024) and PepGLAD (Kong et al., 2024), while PepFlow (Li et al., 2024) and PPFlow (Lin et al., 2024) use geometric flow matching. Despite their progress, these approaches typically require tens to hundreds of neural function evaluations (NFEs) to generate each candidate. This repeated computation limits sampling efficiency, motivating methods that preserve sequence–structure quality with substantially fewer evaluations.

Flow Maps provide a route to few-step generation by learning finite-time transitions rather than repeatedly integrating local dynamics (Boffi et al., 2024; 2025). Extensions to Riemannian and categorical domains support transport of geometric and sequence-valued variables (Davis et al., 2026; Potaptchik et al., 2026), and recent work has explored few-step protein backbone generation (Woo et al., 2026; Bai et al., 2026). Applying finite-time transport to pocket-conditioned peptide co-design, however, requires more than accurate updates of individual modalities. Sequence, residue

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.

positions, and backbone orientations jointly determine the peptide and its receptor interactions (Li et al., 2024); their evolution must remain coordinated over each finite interval.

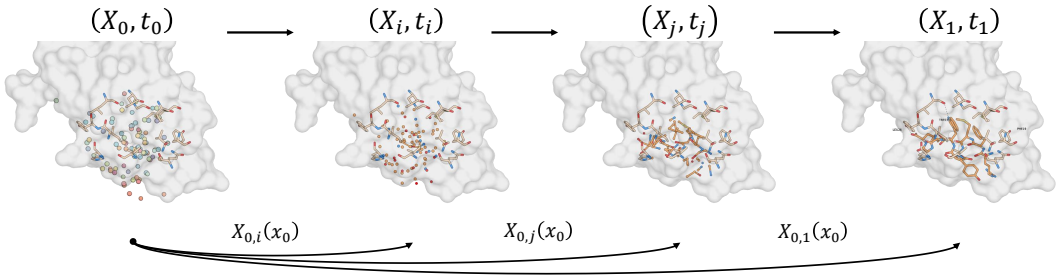


Figure 1: Few-step generation via finite-time transport: $X^{\theta}_{s,t}$ reaches intermediate or bound states in one network evaluation, without iterative integration. Only structure is shown; continuous sequence states evolve synchronously.

This coupling becomes particularly important when a small number of updates spans the full generative trajectory. Local flow-matching supervision alone does not directly enforce agreement between a large-interval transition and a composition of shorter transitions (Boffi et al., 2025). For joint sequence–structure generation, such agreement must account for the complete predicted intermediate state, including dependencies between sequence and geometry. Moreover, transition consistency alone does not ensure plausible backbone geometry or receptor-relative placement. Few-step peptide co-design therefore calls for a joint transport formulation that combines cross-modal coordination with geometric supervision.

We introduce **PepMap**, a pocket-conditioned multimodal Flow Map for few-step peptide sequence–structure co-design with all-atom completion. PepMap jointly transports continuous sequence states, residue positions, and backbone orientations through a shared receptor-conditioned operator (Fig. 1). Building on flow-map self-distillation (Davis et al., 2026), we combine diagonal flow-matching supervision with multimodal semigroup consistency between direct and composed transitions. The composed transition propagates the complete predicted sequence–backbone state, allowing consistency supervision to account for dependencies across modalities. Auxiliary geometric objectives regularize backbone structure and peptide–pocket compatibility. We factorize all-atom generation into this joint sequence–backbone transport followed by conditional side-chain packing, which reconstructs side-chain conformations from the generated state and receptor context.

We evaluate PepMap on the LNR benchmark (Tsaban et al., 2022) across sampling budgets. A single generator evaluation achieves a peptide-aligned L-RMSD of 1.12 Å, outperforming the best standard-budget baseline at 1.48 Å. With ten evaluations, PepMap reaches a C-RMSD of 2.43 Å and an L-RMSD of 1.05 Å, surpassing all evaluated standard-budget baselines on both metrics using only $1/10$–$1/20$ as many generator evaluations. Evaluations of side-chain packing, geometric validity, diversity, and runtime characterize the broader quality–efficiency trade-off. Although few-step structural recovery is strong, interface quality and receptor–peptide steric compatibility remain areas for improvement.

Our contributions are summarized as follows:

- We introduce a pocket-conditioned multimodal Flow Map that enables one- and few-step peptide sequence–backbone co-design, followed by conditional side-chain packing for all-atom completion.
- We formulate joint finite-time transport with geometric and sequence consistency across direct and composed transitions of the complete predicted state, complemented by flow-matching and auxiliary geometric supervision.
- We demonstrate strong structural recovery at substantially reduced sampling budgets on LNR and characterize the resulting trade-offs in interface quality, geometric validity, diversity, and all-atom completion.

2 Related Work

Flow Maps and Few-Step Generation. Few-step generation typically compresses pre-trained samplers by distillation (Salimans & Ho, 2022; Yin et al., 2024; Sauer et al., 2024), or learns direct or self-consistent transitions, as in Consistency Models and related objectives (Song et al., 2023; Song & Dhariwal, 2024; Frans et al., 2025; Zhou et al., 2025). Flow Map Matching and its generalised self-distilled variant G-PSD (Boffi et al., 2024; 2025; Davis et al., 2026) instead learn finite-time solution operators from scratch, while MeanFlow (Geng et al., 2025) parameterizes interval-averaged velocities. These ideas extend to Riemannian and discrete domains (Zhong et al., 2026; Roos et al., 2026; Potaptchik et al., 2026) and to biomolecular generation (Ripken et al., 2026; Fan et al., 2026; Woo et al., 2026; Bai et al., 2026; Scarpellini et al., 2026). Building on these developments, PepMap learns pocket-conditioned flow maps for joint peptide sequence–backbone generation.

Pocket-Conditioned Peptide Design. Target-conditioned peptide design ranges from sequence-level models such as PepMLM (Chen et al., 2025a) to pocket-conditioned sequence–structure co-design. DiffPepBuilder (Wang et al., 2024) and PepGLAD (Kong et al., 2024) use diffusion, whereas PepFlow (Li et al., 2024) and PPFlow (Lin et al., 2024) employ geometric flow matching; recent methods further incorporate hotspots, molecular surfaces, cross-domain modeling, or Bayesian multimodal updates (Li et al., 2025; Wu et al., 2025a; Kong et al., 2025; Qian et al., 2026). However, these approaches predominantly rely on iterative sampling. PepMap instead learns finite-time transport of a coupled sequence–backbone state for few-step pocket-conditioned co-design.

All-Atom Generation and Side-chain Modeling. All-atom generative models either jointly generate sequence, backbone, and side chains, as in Protpardelle (Chu et al., 2024) and recent peptide models (Li et al., 2024; Kong et al., 2024; Qian et al., 2026), or factorize sequence–backbone generation from side-chain reconstruction, as in APM (Chen et al., 2025b). Dedicated side-chain packers range from rotamer-based optimization to deep geometric and generative torsion models (Krivov et al., 2009; McPartlon & Xu, 2023; Visani et al., 2024; Zhang et al., 2023; Lee & Kim, 2025). PepMap adopts the factorized formulation, restricting few-step finite-time transport to the coupled sequence–backbone state and completing side chains with a pocket-conditioned packer.

3 Background and Preliminaries

Factorized All-Atom Peptide Representation We represent a peptide by amino-acid types, backbone frames, and side-chain torsion angles (Fisher, 2001). Residue i has type $a_i \in \{1, \dots, 20\}$, a backbone frame given by the C_α position $x_i \in \mathbb{R}^3$ and an orientation $R_i \in \mathrm{SO}(3)$ built from the N–C_α–C frame (Jumper et al., 2021), and up to four side-chain torsions $\chi_i \in [0, 2\pi)^4$. Given receptor context C_{rec}, all-atom peptide design models $p(C_{\mathrm{pep}} \mid C_{\mathrm{rec}})$ with $C_{\mathrm{pep}} = (x, R, a, \chi)$.

Generating sequence, backbone, and side chains jointly is natural, but prior work has identified limitations. ProBayes (Wu et al., 2025b) shows that side-chain geometry can act as an *information shortcut* for sequence prediction; APM (Chen et al., 2025b) finds that jointly noised side chains may retain residue-identity cues, causing *sequence leakage* and *train–inference mismatch*; and NS-Pep (Guo et al., 2025) notes that side-chain topology is ambiguous before identities are resolved, potentially causing *inconsistent generation*.

Motivated by these observations, we adopt the factorized formulation

$$p(C_{\mathrm{pep}} \mid C_{\mathrm{rec}}) = p(x, R, a \mid C_{\mathrm{rec}})\, p(\chi \mid x, R, a, C_{\mathrm{rec}}). \tag{1}$$

This factorization motivates a two-stage architecture: sequence and backbone are generated jointly under pocket conditioning, followed by conditional side-chain reconstruction. In the implementation, endpoint sequence probabilities provide the packer's sequence features, while decoded residue identities determine atom types for reconstruction.

From Flow Matching to Flow Maps Flow matching (Lipman et al., 2022; Chen & Lipman, 2024) learns a velocity field for a prescribed probability path. In Euclidean space, independently sample $x_0 \sim p_0$, $x_1 \sim p_1$, and $t \sim \mathcal{U}(0, 1)$, and define the linear training interpolant and conditional FM

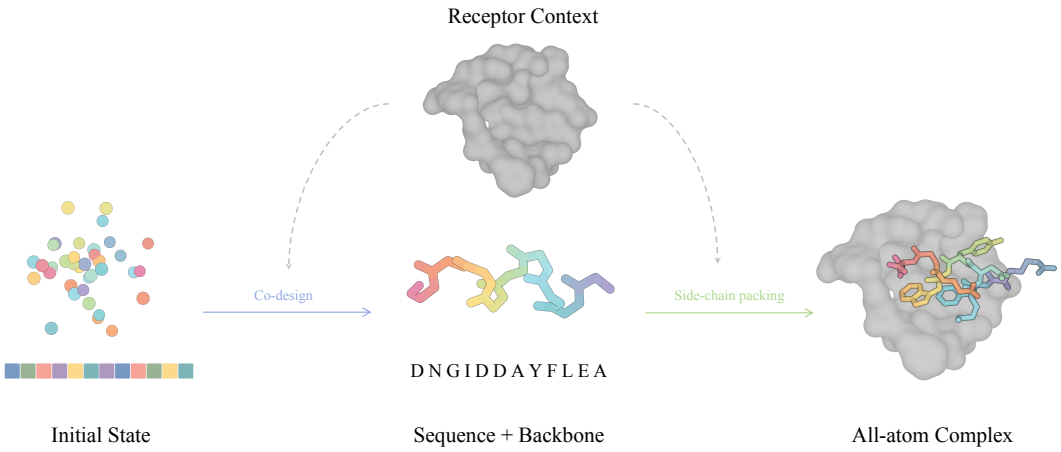


Figure 2: Overview of PepMap's factorized design pipeline. A pocket-conditioned multimodal Flow Map jointly generates peptide sequence and bound backbone in one or few steps, followed by conditional side-chain packing to produce the all-atom receptor–peptide complex.

objective:

$$I_t = (1-t)x_0 + tx_1,$$
$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x_0,x_1,t}\left[\|v_\theta(I_t,t) - (x_1 - x_0)\|_2^2\right]. \tag{2}$$

Over unrestricted predictors, the population minimizer is the marginal velocity field b_t, which defines the generation ODE:

$$b_t(x) = \mathbb{E}[x_1 - x_0 \mid I_t = x], \qquad \dot{y}_t = b_t(y_t), \quad y_0 \sim p_0. \tag{3}$$

Under suitable regularity conditions, this ODE has the same time marginals as I_t, although its trajectories need not coincide with individual training interpolants.

The exact flow map (Boffi et al., 2024; 2025) is the finite-time solution operator of this ODE. For $s \le r \le t$ on intervals where the flow is well-defined,

$$X_{s,t}(y_s) = y_t, \qquad X_{t,t} = \mathrm{Id}, \qquad X_{s,t} = X_{r,t} \circ X_{s,r}. \tag{4}$$

The last identity expresses semigroup consistency. A learned approximation $X^\theta_{s,t}$ is trained to respect these properties; compositional consistency is not automatic. A general Euclidean parameterization uses interval-averaged velocity (Geng et al., 2025); for the exact map, this average follows the ODE trajectory:

$$X^\theta_{s,t}(y) = y + (t-s)u^\theta_{s,t}(y),$$
$$u_{s,t}(y) = \frac{1}{t-s}\int_s^t b_\tau(X_{s,\tau}(y))\, d\tau, \qquad t > s. \tag{5}$$

If the drift is continuous along the trajectory, $\lim_{t\downarrow s} u_{s,t}(y) = b_s(y)$. PepMap instantiates finite-time transitions through the endpoint parameterization in Sec. 4.1.

4 Method

PepMap (Fig. 2) learns pocket-conditioned finite-time transport of a coupled sequence–backbone state using diagonal data supervision and off-diagonal self-distillation. A conditional packer then reconstructs side chains from the generated sequence and backbone.

4.1 Pocket-Conditioned Multimodal Flow Map

Given a receptor pocket and peptide length N, we transport the joint state

$$Z_t = (x_t, R_t, z_t) \in \mathcal{Z} := \mathbb{R}^{N\times 3} \times \mathrm{SO}(3)^N \times \mathbb{R}^{N\times 20}, \tag{6}$$

where z_t is a continuous sequence state without probability normalization and the receptor context C_{rec} remains fixed. The native endpoint has $z_1 = \text{onehot}(a_1)$. We initialize positions with centered Gaussian noise in pocket coordinates, rotations uniformly on $\text{SO}(3)$, and sequence states from standard Gaussian noise. Training uses linear interpolants for positions and sequences and geodesic interpolation for rotations (Appendix A.2).

A shared network predicts interval-dependent endpoints from the joint state, receptor, and both times:

$$\widehat{Z}_{1;s,t} = f_\theta(Z_s, s, t, C_{\text{rec}}) = (\hat{x}_{1;s,t}, \hat{R}_{1;s,t}, \hat{p}_{1;s,t}). \tag{7}$$

The interval-dependent endpoint parameter defines transport from s to t and is returned by the map when $t = 1$. Peptide and receptor features share single and pair representations, refined by a Pairformer (Abramson et al., 2024; Yu et al., 2026) and a two-time-conditioned IPA trunk that updates only peptide frames. Translation, rotation, and sequence heads decode this shared representation; Softmax converts sequence logits to $\hat{p}_{1;s,t} \in (\Delta^{19})^N$. All heads are thus conditioned on the joint state and receptor context. Architectural details appear in Appendix C.

Endpoint-Parameterized Transport. For $0 \le s \le t \le 1$ with $s < 1$, define $\alpha_{s,t} = (t - s)/(1 - s)$. The predicted endpoint parameter induces

$$\begin{aligned} X^{x,\theta}_{s,t}(Z_s) &= x_s + \alpha_{s,t}(\hat{x}_{1;s,t} - x_s), \\ X^{R,\theta}_{s,t}(Z_s) &= \text{Exp}_{R_s}\left[\alpha_{s,t}\,\text{Log}_{R_s}(\hat{R}_{1;s,t})\right], \\ X^{z,\theta}_{s,t}(Z_s) &= z_s + \alpha_{s,t}(\hat{p}_{1;s,t} - z_s). \end{aligned} \tag{8}$$

The map preserves the state spaces and satisfies $X^\theta_{s,s} = \text{Id}$ for $s < 1$; we define $X^\theta_{1,1} = \text{Id}$. At $t = 1$, it returns $\widehat{Z}_{1;s,1}$ in one evaluation. Endpoint predictions may vary across intervals, so the parameterization does not enforce $X^\theta_{r,t} \circ X^\theta_{s,r} = X^\theta_{s,t}$ for $s \le r \le t$ (Lemma 2). For affine modalities, consistency requires the direct endpoint to match the composition's weighted equivalent endpoint, although individual predictions may differ. For rotations, we match the direct and composed transported orientations (Appendix B.2). The following objective encourages these consistency relations.

4.2 Learning Finite-Time Transport

Diagonal supervision anchors local generation directions; off-diagonal self-distillation trains finite-interval compositional consistency, following progressive self-distillation and its Riemannian generalization (Boffi et al., 2025; Davis et al., 2026). The generator learns from random initialization without a pretrained teacher.

Data-Anchored Local Directions. At $s = t = \tau$, the map is the identity but its endpoint predictor defines a local direction. The diagonal objective combines three modality losses:

$$\mathcal{L}_{\text{diag}} = \lambda^{\text{FM}}_x \mathcal{L}^{\text{FM}}_x + \lambda^{\text{FM}}_R \mathcal{L}^{\text{FM}}_R + \lambda^{\text{FM}}_p \mathcal{L}^{\text{FM}}_p. \tag{9}$$

Here $\mathcal{L}^{\text{FM}}_x$ and $\mathcal{L}^{\text{FM}}_R$ match endpoint-induced translation and rotation velocities with bounded time weight $w_{\text{time}}(\tau) = \max(1 - \tau, \delta)^{-2}$, as detailed in Appendix A.3 (Yim et al., 2023). The sequence loss is $\mathcal{L}^{\text{FM}}_p = \mathbb{E}[\text{CE}(a_1, \hat{p}_{1;\tau,\tau})]$, averaged over residues. Following the CE denoiser interpretation in Discrete Flow Maps (Potaptchik et al., 2026), the population-optimal sequence predictor is the conditional mean one-hot endpoint given the joint state, time, and receptor context. Through the endpoint-induced velocity, this predictor recovers the local sequence drift. Appendix B.1 provides the derivation under our joint conditioning.

The auxiliary loss $\mathcal{L}_{\text{aux}}$ comprises backbone atom reconstruction, intrapeptide backbone distances, contact-weighted peptide–pocket Cα distances, and Cα repulsion (Appendix A.4). Each term shares $w_{\text{time}}(\tau)$ and activates when $\tau > \tau_g$; together they give

$$\mathcal{L}_{\text{anchor}} = \mathcal{L}_{\text{diag}} + \mathcal{L}_{\text{aux}} \tag{10}$$

Direct and Composed Transitions. For an interpolated state Z_s and $s < u < t$, we compare a direct update with two composed updates:

$$Z^{\text{dir}}_t = X^\theta_{s,t}(Z_s), \quad Z^\theta_u = X^\theta_{s,u}(Z_s), \quad Z^{\text{cmp}}_t = X^\theta_{u,t}(Z^\theta_u). \tag{11}$$

Both branches share the network, and the second composed update receives the complete predicted state $Z_u^\theta = (x_u^\theta, R_u^\theta, z_u^\theta)$ and receptor context, not a data-interpolated state at u. The composed branch supplies stop-gradient targets. Translation and rotation discrepancies are evaluated at the common destination time t, with normalization over the N valid generated residues:

$$\ell_x^{\mathrm{SG}} = \frac{1}{3N} \sum_i \left\| x_{t,i}^{\mathrm{dir}} - \mathrm{sg}[x_{t,i}^{\mathrm{cmp}}] \right\|_2^2, \qquad \ell_R^{\mathrm{SG}} = \frac{1}{3N} \sum_i d_{\mathrm{SO}(3)} \left(R_{t,i}^{\mathrm{dir}}, \mathrm{sg}[R_{t,i}^{\mathrm{cmp}}] \right)^2. \tag{12}$$

Categorical Semigroup Consistency. The sequence branch follows the same goal, $z_t^{\mathrm{dir}} = z_t^{\mathrm{cmp}}$. To supervise this agreement through the endpoint prediction, we rewrite the composed affine update as

$$z_t^{\mathrm{cmp}} = (1 - \alpha_{s,t}) z_s + \alpha_{s,t} \widetilde{p}_{1;s,t},$$

where the equivalent endpoint is

$$\widetilde{p}_{1;s,t} = \gamma_{s,u,t} \hat{p}_{1;s,u} + (1 - \gamma_{s,u,t}) \hat{p}_{1;u,t}(Z_u^\theta), \tag{13}$$

$$\gamma_{s,u,t} = \frac{(1-t)(u-s)}{(1-u)(t-s)} \in [0,1]. \tag{14}$$

Affine composition determines $\gamma_{s,u,t}$ (Eq. 28). Since the direct update has the same form and $\alpha_{s,t} > 0$ for $s < t$, endpoint equality is equivalent to equality of the transported continuous sequence states. We supervise the direct endpoint using the detached equivalent endpoint as a teacher, averaging the teacher-to-student KL over valid residues:

$$\ell_p^{\mathrm{SG}} = \frac{1}{N} \sum_i D_{\mathrm{KL}}(\mathrm{sg}[\widetilde{p}_{1;s,t,i}] \,\|\, \hat{p}_{1;s,t,i}). \tag{15}$$

We combine the three discrepancies into $\mathcal{L}_{\mathrm{SG}}$ using fixed modality coefficients and detached adaptive weights applied only to SG losses (Appendix A.3). The generator objective is

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{anchor}} + \lambda_{\mathrm{SG}} \mathcal{L}_{\mathrm{SG}}, \tag{16}$$

where $\mathcal{L}_{\mathrm{anchor}} = \mathcal{L}_{\mathrm{diag}} + \mathcal{L}_{\mathrm{aux}}$.

4.3 Conditional All-Atom Completion and Sampling

Conditional Side-Chain Completion. After sequence–backbone transport, an IPA-based packer predicts side-chain torsions from the generated backbone, endpoint sequence probabilities, and receptor context:

$$\hat{\chi} = g_\phi(C_{\mathrm{rec}}, x_1^\theta, R_1^\theta, p_1^\theta), \qquad p_1^\theta = z_1^\theta. \tag{17}$$

Following APM's distinction between packing and torsion reconstruction (Chen et al., 2025b), training uses native or detached predicted conditions selected by the diagonal noise time. Both branches reconstruct native torsions with angle-norm regularization; native conditions additionally receive local side-chain FAPE supervision (Jumper et al., 2021). The generator and packer train concurrently, with packing gradients restricted to ϕ. Conditioning and loss definitions are given in Appendix A.5, and the packing architecture in Appendix C.4.

Few-step Inference. Algorithm 1 applies the Flow Map on the uniform schedule $t_k = k/K$, using exactly K generator evaluations; $K = 1$ applies $X_{0,1}^\theta$ directly. Intermediate sequence states remain continuous. Residue identities are decoded only after all transitions; one packing pass then completes the output. The last transition already returns endpoint probabilities,so no additional generator call is needed. Packing is excluded from the Flow Map NFE but included in wall-clock time. Exact semigroup consistency ensures partition invariance (Corollary 1); finite training need not ensure it.

Algorithm 1 Few-step inference with PepMap

Require: Pocket C_{rec}, peptide length N, Flow Map NFEs K

1: Sample $Z_0 = (x_0, R_0, z_0)$ from the base distribution
2: Set $t_k = k/K$ for $k = 0, \ldots, K$
3: **for** $k = 0, \ldots, K-1$ **do**
4: $\quad Z_{t_{k+1}} \leftarrow X_{t_k, t_{k+1}}^\theta(Z_{t_k}; C_{\mathrm{rec}})$
5: **end for**
6: Set $p_1^\theta \leftarrow z_1^\theta$ (final endpoint probabilities)
7: Decode $a_1^\theta \leftarrow \arg\max p_1^\theta$
8: Pack once: $\hat{\chi} \leftarrow g_\phi(C_{\mathrm{rec}}, x_1^\theta, R_1^\theta, p_1^\theta)$
9: Reconstruct and return $C_{\mathrm{pep}}^{\text{all-atom}}$

5 EXPERIMENTS

This section evaluates PepMap on two complementary tasks: *sequence–structure co-design*, which jointly generates a peptide sequence and bound backbone conditioned on a receptor site, and *side-chain packing*, which predicts peptide side-chain torsions given the bound backbone and receptor context. We organize the experiments into **co-design** (Sec. 5.1), **packing** (Sec. 5.2), and **ablation studies** (Sec. 5.3). The experiments address four questions: **Q1**, whether few-step inference achieves high-quality pocket-conditioned peptide design; **Q2**, whether it preserves molecular validity and diversity; **Q3**, whether the factorized side-chain module retains atomistic accuracy; and **Q4**, which design choices enable accurate large-step generation.

Datasets. We train PepMap on PepBench (Kong et al., 2024), which contains 6,105 non-redundant protein–peptide complexes and follows its standard preprocessing and data-splitting protocol. To evaluate generalization to unseen targets, we test on the Large Non-Redundant (LNR) benchmark (Tsaban et al., 2022), which comprises 93 expert-curated protein–peptide complexes. We follow the evaluation protocol of prior work (Kong et al., 2024) to ensure comparability with existing peptide generation methods. Additional dataset details are provided in Appendix D.1.

Table 1: Sequence–structure recovery and binding quality across sampling budgets (NFE).

Model	NFE ↓	Recovery			Binding Quality		
		AAR ↑	C-RMSD ↓	L-RMSD ↓	ΔG ↓	IMP ↑	DockQ ↑
Reference	–	–	–	–	-37.25	–	–
RFDiffusion	200	34.68%	4.69	1.88	-13.47	5.38%	0.481
DiffPepBuilder	200	32.15%	4.87	1.53	-19.72	13.97%	0.535
PepFlow	200	33.86%	4.14	1.56	-21.75	20.43%	0.566
PepGLAD	100	<u>38.62%</u>	2.74	1.60	-23.12	18.28%	0.686
UniMoMo$_{\text{single}}$	100	37.58%	2.48	1.48	**-28.72**	**29.03%**	<u>0.788</u>
PepMap	1	37.89%	2.52	1.12	-22.76	19.35%	0.735
PepMap	5	38.48%	2.45	1.10	-25.82	23.59%	0.743
PepMap	10	38.55%	<u>2.43</u>	<u>1.05</u>	-26.23	24.72%	0.762
PepMap	100	**39.69%**	**2.37**	**0.97**	<u>-27.56</u>	<u>28.18%</u>	**0.792**

Baselines. For sequence–structure co-design, we compare PepMap with five representative methods: **RFDiffusion** (Watson et al., 2023), **DiffPepBuilder** (Wang et al., 2024), **PepGLAD** (Kong et al., 2024), **PepFlow** (Li et al., 2024), and **UniMoMo$_{\text{single}}$** (Kong et al., 2025), covering diffusion- and flow-based generative paradigms. DiffPepBuilder and PepFlow are retrained on the training split used in this study. For PepFlow co-design, we replace the native peptide centroid with the receptor pocket centroid as the initialization center. UniMoMo uses the peptide-only UniMoMo$_{\text{single}}$ variant. For side-chain packing, we compare PepMap only with PepFlow under native sequence and backbone conditions, with the receptor held fixed. Further implementation details are provided in Appendix D.2.

5.1 SEQUENCE–STRUCTURE CO-DESIGN

Metrics. We evaluate sequence–structure co-design from multiple perspectives, including sequence recovery, structural quality, binding capability, physical plausibility, generative diversity, and sampling efficiency. Specifically, **AAR** (Needleman & Wunsch, 1970; Henikoff & Henikoff, 1992), **C-RMSD**, and **L-RMSD** measure sequence and structural recovery with respect to the reference complexes, while **DockQ** (Mirabello & Wallner, 2024), pyRosetta (Alford et al., 2017) binding energy $\boldsymbol{\Delta G}$, and **IMP** assess binding and interface quality. Physical plausibility is evaluated using $\textbf{Clash}_{\textbf{in}}$ and $\textbf{Clash}_{\textbf{out}}$, which characterize steric conflicts within the peptide and across the peptide–receptor interface, together with $\textbf{JSD}_{\textbf{bb}}$ and $\textbf{JSD}_{\textbf{sc}}$ for backbone and side-chain torsional distributions. We additionally report **sequence** and **structural diversity** to characterize the coverage of the generated samples. Sampling efficiency is measured by **NFE** and **wall-clock time**. Detailed metric definitions and evaluation protocols are provided in Appendix D.3.

Few-step Generation Quality (Q1). Table 1 compares PepMap with standard-budget baselines across inference budgets. At 1 NFE, PepMap outperforms all standard-budget baselines on L-RMSD (1.12 Å) and nearly matches the best C-RMSD (2.52 vs. 2.48 Å), although its DockQ is lower (0.735 vs. 0.788). At $K = 10$, PepMap surpasses all standard-budget baselines on C-RMSD and L-RMSD (2.43 Å and 1.05 Å) using $1/10$–$1/20$ as many NFEs. DockQ reaches 0.762, but UniMoMo$_{\text{single}}$ leads on DockQ (0.788), ΔG, and IMP. At $K = 100$, PepMap leads in AAR, C-RMSD, L-RMSD, and DockQ, while UniMoMo$_{\text{single}}$ retains better ΔG (-28.72 vs. -27.56 REU) and IMP (29.03% vs. 28.18%). PepMap uses diagonal-field Euler sampling (Appendix A.6), changing both budget and sampler. Figure 3 shows robust PepMap performance at low budgets, whereas iterative baselines degrade markedly. Matched-budget comparisons at 1 and 10 NFE are provided in Appendix E.1, where one-step PepMap outperforms four 10-NFE baselines on C-RMSD, L-RMSD, and DockQ. Its one-step L-RMSD also surpasses the best full-budget baseline (1.12 vs. 1.48 Å), highlighting the effectiveness of learned finite-time transport for large generative transitions. Figure 4 (top) shows 6QG8 pose recovery with PepMap at 1 and 10 NFE.

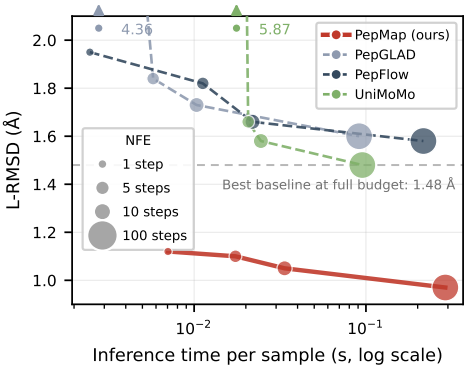


Figure 3: L-RMSD versus per-sample inference time on LNR (log scale). Marker size indicates NFE; the dashed line marks the best full-budget baseline (1.48 Å).

Table 2: Physical validity and sequence–structure diversity across sampling budgets (NFE).

Model	NFE ↓	Rationality				Diversity	
		Clash$_{\text{in}}$ ↓	Clash$_{\text{out}}$ ↓	JSD$_{\text{bb}}$ ↓	JSD$_{\text{sc}}$ ↓	Seq. ↑	Struct. ↑
Reference	–	0.31%	0.88%	–	–	–	–
RFDiffusion	200	**0.06%**	13.58%	0.273	0.798	0.155	0.616
DiffPepBuilder	200	2.84%	12.44%	0.238	0.358	0.015	0.077
PepFlow	200	1.97%	7.34%	0.247	0.694	0.043	0.253
PepGLAD	100	1.82%	<u>1.66%</u>	0.474	0.398	**0.687**	**0.698**
UniMoMo$_{\text{single}}$	100	1.53%	**0.94%**	0.390	0.365	0.626	<u>0.629</u>
PepMap	1	4.12%	9.41%	0.233	0.292	<u>0.683</u>	0.607
PepMap	5	0.77%	6.53%	0.150	0.289	0.656	0.580
PepMap	10	<u>0.62%</u>	6.16%	<u>0.144</u>	<u>0.288</u>	0.648	0.576
PepMap	100	**0.06%**	4.37%	**0.141**	**0.285**	0.580	0.272

Physical Validity and Diversity (Q2). Table 2 shows that PepMap maintains competitive torsional distributions and diversity under few-step sampling. From $K = 1$ to $K = 10$, Clash$_{\text{in}}$ decreases from 4.12% to 0.62% and JSD$_{\text{bb}}$ from 0.233 to 0.144, while JSD$_{\text{sc}}$ remains stable (0.292–0.288). At $K = 10$, both JSDs are lower than those of all evaluated baselines, with competitive sequence and structural diversity (0.648 and 0.576). These results show favorable torsional statistics despite separating side-chain packing from the sequence–backbone finite-time transport process. However, receptor–peptide Cα Clash$_{\text{out}}$ decreases from 9.41% to 6.16%, above PepGLAD (1.66%) and UniMoMo$_{\text{single}}$ (0.94%) at standard sampling budgets. These baselines use additional protein-fragment data (Kong et al., 2024; 2025), whereas PepMap uses only PepBench. Data and reconstruction differences may contribute to this gap, but require controlled evaluation. The single-step 6J0X example shows fewer peptide clashes than the displayed baselines (Figure 4, bottom).

5.2 Side-Chain Packing

Metrics. We evaluate the quality of generated side-chain conformations using two metrics: (1) **Angle MAE** measures the mean absolute error of predicted torsional angles, reported in degrees for χ_1–χ_4. (2) **Angle Accuracy** measures the proportion of correct predictions among valid torsional angles, considering an angle correct if its deviation from the native value is within 20°.

Table 3: Packing results on LNR dataset.

Method	Angle MAE ° ↓				Angle Accuracy (%) ↑
	χ_1	χ_2	χ_3	χ_4	
PepFlow	73.08	79.41	93.13	115.48	8.64
PepMap	58.17	46.89	65.57	69.52	16.62

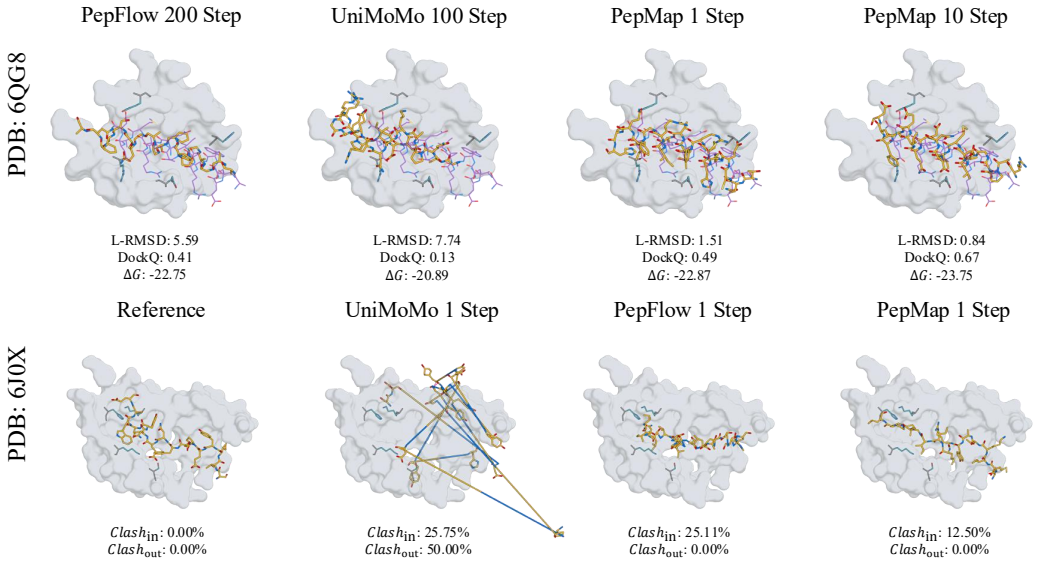


Figure 4: **Top (Q1)**: superposition of the native peptide and samples generated at the indicated NFE. **Bottom (Q2)**: reference and single-step outputs for 6J0X, with per-structure clash values.

Side-chain Packing Accuracy (Q3). Under matched native sequence, backbone, and receptor conditions, PepMap reduces MAE for all four side-chain torsions relative to PepFlow and improves Angle Accuracy from 8.64% to 16.62% (Table 3). Rosetta more accurately recovers native rotamers in this setting. On PepMap-generated inputs, however, PepMap's packer better matches reference side-chain torsional distributions, while Rosetta yields more favorable post-relaxation ΔG and IMP (Appendix E.2). Together, these results distinguish native rotamer recovery, distributional plausibility, and post-relaxation interface energetics.

5.3 Ablation Studies

Component Contributions (Q4). Table 4 compares three component ablations on LNR at $K = 1$ and $K = 10$. Removing semigroup supervision yields the largest losses in structural recovery and interface quality, even with ten sampling steps. Its ten-step results remain worse than full-model single-step results on both metrics. Pairformer improves all reported metrics at both budgets, while geometric supervision primarily benefits pose recovery and interface compatibility. However, removing geometric supervision slightly reduces internal clashes at $K = 1$, indicating that its benefits vary across metrics. Detailed settings and metric-specific effects are discussed in Appendix E.3.

Table 4: Co-design ablations on LNR at $K = 1$ and $K = 10$.

	$K = 1$					$K = 10$				
	C-RMSD ↓	DockQ ↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓	JSD_{bb} ↓	C-RMSD ↓	DockQ ↑	Clash_{in} ↓	$\text{Clash}_{\text{out}}$ ↓	JSD_{bb} ↓
PepMap	**2.52**	**0.735**	4.12%	**9.41%**	**0.233**	**2.43**	**0.762**	**0.62%**	**6.16%**	**0.144**
w/o Semigroup	5.21	0.513	17.02%	14.09%	0.391	5.01	0.534	7.50%	12.8%	0.246
w/o Pairformer	3.00	0.657	7.26%	10.87%	0.279	2.96	0.688	0.89%	7.48%	0.177
w/o Geometry	4.01	0.626	**3.71%**	16.25%	0.242	3.60	0.638	1.57%	14.11%	0.149

6 Conclusion

We introduced **PepMap**, a pocket-conditioned multimodal Flow Map for few-step peptide sequence–structure co-design with all-atom completion. By learning finite-time transitions of the coupled sequence–backbone state, PepMap achieves strong structural recovery with one to ten generator evaluations, while conditional side-chain packing reconstructs the remaining atomistic detail. Results on LNR show that accurate peptide co-design is possible at substantially reduced sampling budgets, supporting multimodal finite-time transport as an efficient alternative to iterative generation. Improving receptor–peptide steric compatibility and experimentally validating generated candidates remain important directions for future work.

AI USE STATEMENT

We used generative AI tools to assist with code implementation. We also used an AI assistant to check submission requirements and manuscript formatting, migrate the manuscript to the ICLR 2027 template, and draft the AI use, ethics, and reproducibility statements. The authors are responsible for the final content of this work, including AI-assisted text and code.

ETHICS STATEMENT

Efficient peptide binder design may support biological research and therapeutic discovery. The ability to generate peptides conditioned on a protein target also carries potential for misuse, including the design of molecules with harmful biological effects. Downstream use should therefore consider the biological function of the target and applicable biosafety requirements. The computational evaluations reported here assess generated structures and related metrics; experimental validation of binding, specificity, and safety is needed before biological or therapeutic use.

REPRODUCIBILITY STATEMENT

Appendix A specifies the representation, priors, training objectives, and inference procedures, and Appendix B provides the mathematical derivations. Model architecture details are given in Appendix C. Appendix D describes the datasets and splits, baselines, evaluation metrics, training settings, and runtime measurement protocol. These details include the per-target candidate selection protocol and the sampling settings used for the reported comparisons. Additional comparisons and ablations are provided in Appendix E, with structural visualizations across sampling budgets in Appendix F.

REFERENCES

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

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

Yikun Bai, Binghang Lu, Yikai Liu, Elaheh Akbari, Soheil Kolouri, Linxuan Wang, Ping He, Shuchan Wang, Ruqi Zhang, and Guang Lin. Se (3)-meanflow: Few-step protein backbone generation on lie groups. *arXiv preprint arXiv:2607.27431*, 2026.

Gaurav Bhardwaj, Vikram Khipple Mulligan, Christopher D Bahl, Jason M Gilmore, Peta J Harvey, Olivier Cheneval, Garry W Buchko, Surya VSRK Pulavarti, Quentin Kaas, Alexander Eletsky, et al. Accurate de novo design of hyperstable constrained peptides. *Nature*, 538(7625):329–335, 2016.

Nicholas Boffi, Michael Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *Advances in Neural Information Processing Systems*, 38: 33346–33382, 2025.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507*, 2024.

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

Leo Tianlai Chen, Zachary Quinn, Madeleine Dumas, Christina Peng, Lauren Hong, Moises Lopez-Gonzalez, Alexander Mestre, Rio Watson, Sophia Vincoff, Lin Zhao, et al. Target sequence-conditioned design of peptide binders using masked language modeling. *Nature Biotechnology*, pp. 1–9, 2025a.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, volume 2024, pp. 47922–47945, 2024.

Ruizhe Chen, Dongyu Xue, Xiangxin Zhou, Zaixiang Zheng, Xiangxiang Zeng, and Quanquan Gu. An all-atom generative model for designing protein complexes. *arXiv preprint arXiv:2504.13075*, 2025b.

Chaoran Cheng, Zhanghan Ni, Yanru Qu, Yuxin Chen, Ruihan Guo, Jiajun Fan, and Ge Liu. Variable-length generative protein design via generalized poisson flow. *arXiv preprint arXiv:2607.09039*, 2026.

Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

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

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. Ragotte, L. Milles, B. Wicky, A. Courbet, R. de Haas, N. Bethel, P. Leung, T. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. Bera, N. King, and D. Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

Oscar Davis, Michael Albergo, Nicholas Boffi, Michael Bronstein, and Joey Bose. Generalised flow maps for few-step generative modelling on riemannian manifolds. In *International Conference on Learning Representations*, volume 2026, pp. 31119–31140, 2026.

Roland L Dunbrack, Jr. and Fred E Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6(8):1661–1681, 1997. doi: 10.1002/pro.5560060807.

Zhiguang Fan, Yuedong Yang, Mingyuan Xu, and Hongming Chen. Ec-dock: A fast equivariant consistency model for molecular docking and virtual screening. *Journal of Chemical Information and Modeling*, 2026.

Matthew Fisher. Lehninger principles of biochemistry, by david l. nelson and michael m. cox. *The Chemical Educator*, 6(1):69–70, 2001.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-Proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2025.

Tao Guo, Junbo Yin, Yu Wang, and Xin Gao. Ns-pep: De novo peptide design with non-standard amino acids. *arXiv preprint arXiv:2510.03326*, 2025.

Erik Hartman, Jonas Wallin, Johan Malmström, and Jimmy Olsson. Controllable protein design through Feynman–Kac steering. *arXiv preprint arXiv:2511.09216*, 2025.

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.

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

Jerry Y. Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M. Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026.

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

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

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

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

Georgii G Krivov, Maxim V Shapovalov, and Roland L Dunbrack Jr. Improved prediction of protein side-chain conformations with scwrl4. *Proteins: Structure, Function, and Bioinformatics*, 77(4): 778–795, 2009.

Jin Sub Lee and Philip M Kim. FlowPacker: Protein side-chain packing with torsional flow matching. *Bioinformatics*, 41(3):btaf010, 2025.

Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-atom peptide design based on multi-modal flow matching. In *International Conference on Machine Learning*, pp. 27615–27640. PMLR, 2024.

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*, volume 2025, pp. 9282–9314, 2025.

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 *International Conference on Machine Learning*, pp. 30510–30528. PMLR, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Matthew McPartlon and Jinbo Xu. An end-to-end deep learning method for protein side-chain packing and inverse folding. *Proceedings of the National Academy of Sciences*, 120(23):e2216438120, 2023.

Claudio Mirabello and Björn Wallner. Dockq v2: improved automatic quality measure for protein multimers, nucleic acids, and small molecules. *Bioinformatics*, 40(10):btae586, 2024.

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.

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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S Albergo. Discrete flow maps. *arXiv preprint arXiv:2604.09784*, 2026.

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

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

Winfried Ripken, Michael Plainer, Gregor Lied, Thorben Frank, Oliver T Unke, Stefan Chmiela, Frank Noé, and Klaus-Robert Müller. Learning hamiltonian flow maps: Mean flow consistency for large-timestep molecular dynamics. *arXiv preprint arXiv:2601.22123*, 2026.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, Ismail Ilkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Gianluca Scarpellini, Ron Shprints, Peter Holderrieth, Juno Nam, Pranav Murugan, Rafael Gómez-Bombarelli, Tommi Jaakkola, Maruan Al-Shedivat, Nicholas Matthew Boffi, and Avishek Joey Bose. Few-step cofolding with all-atom flow maps. *arXiv preprint arXiv:2606.08375*, 2026.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

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

Gian Marco Visani, William Galvin, Michael Pun, and Armita Nourmohammad. H-packer: Holographic rotationally equivariant convolutional neural network for protein side-chain packing. In *Machine Learning in Computational Biology*, pp. 230–249. PMLR, 2024.

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.

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

Dongyeop Woo, Marta Skreta, Seonghyun Park, Kirill Neklyudov, and Sungsoo Ahn. Riemannian meanflow. *arXiv preprint arXiv:2602.07744*, 2026.

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

Hanlin Wu, Yuxuan Song, Zhe Zhang, Zhilong Zhang, Hao Zhou, Wei-Ying Ma, and Jingjing Liu. Rationalized all-atom protein design with unified multi-modal bayesian flow. *Advances in Neural Information Processing Systems*, 38:3019–3050, 2025b.

Fei Ye, Zaixiang Zheng, Dongyu Xue, Yuning Shen, Lihao Wang, Yiming Ma, Yan Wang, Xinyou Wang, Xiangxin Zhou, and Quanquan Gu. ProteinBench: A holistic evaluation of protein foundation models. In *International Conference on Learning Representations*, volume 2025, pp. 29857–29891, 2025.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Qilin Yu, Liangyue Guo, Xiayan Qin, Xikun Huang, Baihui Tian, Hongzhun Wang, Yu Liu, Yunzhi Lang, Di Wang, Zhouhanyu Shen, et al. High-affinity protein binder design via flow matching and in silico maturation. *bioRxiv*, pp. 2026–01, 2026.

Yangtian Zhang, Zuobai Zhang, Bozitao Zhong, Sanchit Misra, and Jian Tang. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing. *Advances in Neural Information Processing Systems*, 36:48150–48172, 2023.

Zichen Zhong, Haoliang Sun, Yukun Zhao, Yongshun Gong, and Yilong Yin. Riemannian meanflow for one-step generation on manifolds. *arXiv preprint arXiv:2603.10718*, 2026.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

A Algorithm Details

A.1 Notation and Geometric Conventions

PepMap models the coupled sequence–backbone state

$$Z_t = (x_t, R_t, z_t) \in \mathbb{R}^{N\times 3} \times \mathrm{SO}(3)^N \times \mathbb{R}^{N\times 20},$$

where N is the number of valid peptide residues to be generated, x_t contains Cα positions, R_t contains backbone orientations, and z_t contains continuous sequence states without probability normalization. The native sequence endpoint is $z_1 = \mathrm{onehot}(a_1)$, whereas predicted sequence endpoints $\hat{p}_{1;s,t}$ lie in $(\Delta^{19})^N$, with $\Delta^{19} = \{p \in \mathbb{R}^{20}_{\geq 0} : \mathbf{1}^\top p = 1\}$.

The receptor pocket context C_{rec}, including receptor frames, remains fixed during transport. Sidechain torsions are reconstructed conditionally and are not part of Z_t. Time runs from noise at 0 to data at 1. We use $\tau \in [0, 1)$ for diagonal evaluations with $s = t = \tau$, and $0 \leq s < u < t \leq 1$ for composition through an intermediate time u.

Each residue frame is a rigid transformation $T_i = (R_i, x_i) \in \mathrm{SE}(3)$, mapping local coordinates to pocket coordinates as $T_i(\xi) = R_i\xi + x_i$. For backbone geometry, we use the product of the Euclidean translation metric and the bi-invariant rotation metric specified below. Rotational tangent vectors are represented by body-frame coordinates $\omega \in \mathbb{R}^3$, corresponding to $R[\omega]_\times \in T_R\mathrm{SO}(3)$, where $[\omega]_\times$ is the skew-symmetric matrix representing the cross product with ω. With $\mathrm{Exp}(\omega) = \exp([\omega]_\times)$, we write the exponential and logarithm maps in these coordinates as

$$\mathrm{Exp}_R(\omega) = R\,\mathrm{Exp}(\omega), \qquad \mathrm{Log}_R(R') = \mathrm{Log}(R^\top R'), \qquad d_{\mathrm{SO}(3)}(R, R') = \|\,\mathrm{Log}_R(R')\|_2.$$

Here Log returns a principal-branch rotation vector. For relative angles below π, this determines a unique shortest geodesic. These rotational operations are applied separately to each peptide residue.

A.2 Priors and Flow-Map Parameterization

The receptor pocket defines the spatial reference. Let $\mathcal{I}_{\mathrm{rec}}$ index its valid residues. We express peptide and receptor Cα coordinates relative to the pocket centroid:

$$\begin{aligned} c_{\mathrm{pocket}} &= \frac{1}{|\mathcal{I}_{\mathrm{rec}}|} \sum_{j\in\mathcal{I}_{\mathrm{rec}}} r_j^{\mathrm{abs}}, \\ x_{1,i} = x_{1,i}^{\mathrm{abs}} - c_{\mathrm{pocket}}, &\qquad r_j = r_j^{\mathrm{abs}} - c_{\mathrm{pocket}}. \end{aligned} \tag{18}$$

The native peptide is not separately centered at its own centroid. Absolute coordinates are recovered by adding c_{pocket}.

The translation prior is constructed from centered Gaussian noise:

$$\epsilon_i \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, I_3), \qquad x_{0,i} = \sigma_x \left(\epsilon_i - \frac{1}{N}\sum_{j=1}^{N} \epsilon_j \right), \tag{19}$$

where $\sigma_x > 0$ is the translation noise scale. This gives a zero initial peptide centroid without using native peptide coordinates. Centering correlates the residue translations; the underlying ϵ_i are independent. Rotations and continuous sequence states have priors

$$R_{0,i} \sim \mathcal{U}(\mathrm{SO}(3)), \qquad z_{0,i} \sim \mathcal{N}(0, I_{20}).$$

Given a base state Z_0 and a native state Z_1, we construct training states by linear interpolation of positions and continuous sequence states, and geodesic interpolation of rotations:

$$\begin{aligned} x_\tau &= (1-\tau)x_0 + \tau x_1, \\ R_\tau &= \mathrm{Exp}_{R_0}\left[\tau\,\mathrm{Log}_{R_0}(R_1)\right], \\ z_\tau &= (1-\tau)z_0 + \tau\,\mathrm{onehot}(a_1). \end{aligned} \tag{20}$$

These interpolants construct training states; finite-time transport is parameterized by the learned map below. Training and sampling share the same sequence prior. The predictor consumes continuous sequence states directly and applies softmax only to sequence endpoint logits.

For $0 \le s \le t \le 1$ with $s < 1$, the network predicts an interval-dependent endpoint parameter:

$$\widehat{Z}_{1;s,t} = f_\theta(Z_s, s, t, C_{\mathrm{rec}}) = (\hat{x}_{1;s,t}, \hat{R}_{1;s,t}, \hat{p}_{1;s,t}),$$
$$\alpha_{s,t} = \frac{t-s}{1-s}.$$

This parameter refers to time 1 and defines the transition from s to t. Following the geometric endpoint parameterization of Riemannian MeanFlow (Woo et al., 2026), we define

$$u^x_{s,t} = \frac{\hat{x}_{1;s,t} - x_s}{1-s}, \qquad u^R_{s,t} = \frac{\mathrm{Log}_{R_s}(\hat{R}_{1;s,t})}{1-s}.$$

The resulting joint map is

$$\begin{aligned} X^{x,\theta}_{s,t}(Z_s) &= x_s + (t-s)u^x_{s,t}, \\ X^{R,\theta}_{s,t}(Z_s) &= \mathrm{Exp}_{R_s}\big((t-s)u^R_{s,t}\big), \\ X^{z,\theta}_{s,t}(Z_s) &= (1-\alpha_{s,t})z_s + \alpha_{s,t}\hat{p}_{1;s,t}. \end{aligned} \tag{21}$$

The map is the identity at $t = s$ and returns $\widehat{Z}_{1;s,1}$ at $t = 1$; we set $X^\theta_{1,1} = \mathrm{Id}$. For each transition, $(t-s)u^R_{s,t}$ specifies the geodesic displacement in body-frame coordinates. Endpoint parameters may vary across intervals, so exact composition is not automatic. The affine composition defect is characterized in Lemma 2; rotational consistency is evaluated directly in transported-orientation space (Appendix B.2).

A.3 Flow-Map Training Objectives

We sample $\tau \sim q_{\mathrm{diag}}$ with $\tau < 1$ for diagonal supervision and $(s, u, t) \sim q_{\mathrm{SG}}$ with $0 \le s < u < t \le 1$ for semigroup supervision. The initial training states Z_τ and Z_s are constructed using Eq. equation 20; the sampling distributions are specified in Appendix D.4. All residue averages below include only valid generated peptide residues.

Diagonal supervision. At $s = t = \tau$, the endpoint predictor defines local translation and rotation velocities through the parameterization in Appendix A.2. Following SE(3) flow matching (Yim et al., 2023), we compare these velocities with those of the training interpolants. Define

$$e_x = \frac{1}{3N}\sum_i \|\hat{x}_{1;\tau,\tau,i} - x_{1,i}\|^2, \tag{22}$$

$$e_R = \frac{1}{3N}\sum_i \left\|\mathrm{Log}_{R_{\tau,i}}(\hat{R}_{1;\tau,\tau,i}) - \mathrm{Log}_{R_{\tau,i}}(R_{1,i})\right\|^2. \tag{23}$$

The rotational differences use body-frame coordinates at the same $R_{\tau,i}$. The corresponding squared velocity errors are $e_x/(1-\tau)^2$ and $e_R/(1-\tau)^2$. To bound the weighting near the data endpoint, we use

$$w_{\mathrm{time}}(\tau) = \max(1-\tau, \delta)^{-2}, \qquad \delta > 0. \tag{24}$$

The diagonal objective combines geometric supervision with sequence endpoint cross-entropy:

$$\begin{aligned} \mathcal{L}_{\mathrm{diag}} = \mathbb{E}\Big[& w_{\mathrm{time}}(\tau)\big(\lambda^{\mathrm{FM}}_x s_x^2 e_x + \lambda^{\mathrm{FM}}_R e_R\big) \\ & + \lambda^{\mathrm{FM}}_p \frac{1}{N}\sum_i \mathrm{CE}(a_{1,i}, \hat{p}_{1;\tau,\tau,i})\Big]. \end{aligned} \tag{25}$$

Here s_x scales translation errors independently of the loss weights. Following the CE denoiser interpretation in Discrete Flow Maps (Potaptchik et al., 2026), the population-optimal predictor for the sequence CE term is the conditional mean one-hot endpoint given the joint state, time, and receptor context. Through the endpoint parameterization, this recovers the local sequence drift (Lemma 1).

Semigroup supervision. Following progressive self-distillation and its Riemannian generalization (Boffi et al., 2025; Davis et al., 2026), we train a direct transition against a two-step composition:

$$\begin{aligned} Z_t^{\mathrm{dir}} &= X_{s,t}^{\theta}(Z_s), \\ Z_u^{\theta} &= X_{s,u}^{\theta}(Z_s), \\ Z_t^{\mathrm{cmp}} = X_{u,t}^{\theta}(Z_u^{\theta}), &\qquad \widetilde{Z}_t = \mathrm{sg}[Z_t^{\mathrm{cmp}}], \end{aligned} \tag{26}$$

where sg denotes stop-gradient. Both branches share the network. The second composed transition receives the complete predicted state Z_u^{θ}, rather than a data-interpolated state at u. Translation and rotation are compared at the common destination:

$$\begin{aligned} \ell_x^{\mathrm{SG}} &= \frac{1}{3N}\sum_i \|x_{t,i}^{\mathrm{dir}} - \tilde{x}_{t,i}\|^2, \\ \ell_R^{\mathrm{SG}} &= \frac{1}{3N}\sum_i d_{\mathrm{SO}(3)}(R_{t,i}^{\mathrm{dir}}, \tilde{R}_{t,i})^2. \end{aligned} \tag{27}$$

Unlike diagonal rotation supervision, the rotational SG loss compares transported orientations directly by their geodesic distance.

For sequence consistency, let $A = \hat{p}_{1;s,u}$ and $B = \hat{p}_{1;u,t}(Z_u^{\theta})$ denote the two teacher endpoint predictions. Applying Lemma 2 gives

$$z_t^{\mathrm{cmp}} = (1-\alpha_{s,t})z_s + \alpha_{s,t}\big[\gamma A + (1-\gamma)B\big],$$

where

$$\gamma = \frac{(1-\alpha_{u,t})\alpha_{s,u}}{\alpha_{s,t}} = \frac{(1-t)(u-s)}{(1-u)(t-s)} \in [0,1]. \tag{28}$$

Thus the composition admits a valid categorical equivalent endpoint. We supervise the direct endpoint with a teacher-to-student KL:

$$\begin{aligned} \tilde{p} = \mathrm{sg}[\gamma A + (1-\gamma)B], &\qquad q = \hat{p}_{1;s,t}, \\ \ell_p^{\mathrm{SG}} &= \frac{1}{N}\sum_i D_{\mathrm{KL}}(\tilde{p}_i \| q_i). \end{aligned} \tag{29}$$

Since $z_t^{\mathrm{dir}} - \tilde{z}_t = \alpha_{s,t}(q - \tilde{p})$, endpoint agreement is equivalent to transported-state agreement for $s < t$. KL is applied only to endpoint distributions, not to the intermediate continuous sequence states.

Adaptive weighting and aggregation. Adaptive weighting is applied only to SG losses. For the geometric modalities, we use squared-error-based weights following Riemannian MeanFlow (Woo et al., 2026). For sequence, we follow Discrete Flow Maps (Potaptchik et al., 2026) and compute the weight from the squared probability discrepancy, rather than the KL value. We first form the per-sample quantities

$$\begin{aligned} d_x &= \lambda_x^{\mathrm{SG}} s_x^2 \ell_x^{\mathrm{SG}}, \\ d_R &= \lambda_R^{\mathrm{SG}} \ell_R^{\mathrm{SG}}, \\ d_p &= \frac{1}{N}\sum_i \|q_i - \tilde{p}_i\|_2^2, \end{aligned} \tag{30}$$

using the same translation scale s_x as diagonal supervision. The weights and combined SG objective are

$$w_m = (d_m + c)^{-\rho}, \qquad m \in \{x, R, p\}, \quad c > 0,\ \rho > 0, \tag{31}$$

$$\begin{aligned} \mathcal{L}_{\mathrm{SG}} = \mathbb{E}\Big[&\lambda_x^{\mathrm{SG}} s_x^2\, \mathrm{sg}(w_x)\ell_x^{\mathrm{SG}} \\ &+ \lambda_R^{\mathrm{SG}}\, \mathrm{sg}(w_R)\ell_R^{\mathrm{SG}} + \lambda_p^{\mathrm{SG}}\, \mathrm{sg}(w_p)\ell_p^{\mathrm{SG}}\Big]. \end{aligned} \tag{32}$$

Each weight is computed after residue averaging and is detached during optimization. Diagonal supervision anchors local predictions to data, whereas SG supervision encourages consistency across finite-time transitions. Loss-clipping thresholds are specified in Appendix D.4.

A.4 Auxiliary Geometric Losses

Transport consistency alone does not ensure accurate backbone or interface geometry. We evaluate four auxiliary losses on the backbone reconstructed from the diagonal endpoint prediction $(\hat{x}_{1;\tau,\tau}, \hat{R}_{1;\tau,\tau})$. Let $\mathcal{V}$ denote the valid peptide backbone atoms of types N, Cα, and C, with reconstructed coordinates $\hat{y}_a$ and native reference coordinates $y_a^\star$. All coordinates are expressed in the common pocket coordinate system. The following losses are defined before time weighting.

The backbone coordinate loss supervises atomic placement and local orientation relative to the pocket:

$$\ell_{\mathrm{bb}} = \frac{1}{|\mathcal{V}|} \sum_{a \in \mathcal{V}} \|\hat{y}_a - y_a^\star\|^2. \tag{33}$$

The intrapeptide distance loss complements coordinate supervision by constraining relative geometry independently of the peptide's rigid-body placement:

$$\ell_{\mathrm{pair}} = \frac{1}{|\mathcal{V}|^2} \sum_{a,b \in \mathcal{V}} \left(\|\hat{y}_a - \hat{y}_b\| - \|y_a^\star - y_b^\star\|\right)^2. \tag{34}$$

For interface supervision, let $\hat{x}_i = \hat{x}_{1;\tau,\tau,i}$ and $x_{1,i}$ denote predicted and native peptide Cα coordinates, and let r_j denote fixed pocket Cα coordinates. Define

$$\hat{d}_{ij} = \|\hat{x}_i - r_j\|, \qquad d_{ij}^\star = \|x_{1,i} - r_j\|.$$

We emphasize native contacts using weights determined only by the reference distances. For contact cutoff r_c and transition width $\Delta > 0$,

$$\begin{aligned} v_{ij} &= \mathrm{clip}\left(\frac{r_c + \Delta - d_{ij}^\star}{\Delta}, 0, 1\right), \qquad h_{ij} = v_{ij}^2(3 - 2v_{ij}), \\ \ell_{\mathrm{int}} &= \frac{\sum_{i,j} h_{ij}(\hat{d}_{ij} - d_{ij}^\star)^2}{\sum_{i,j} h_{ij}}. \end{aligned} \tag{35}$$

The weights equal one for $d_{ij}^\star \le r_c$ and decrease smoothly to zero at $r_c + \Delta$. Samples with zero total contact weight contribute zero.

A Cα repulsion term penalizes peptide residues whose nearest pocket Cα lies within r_{clash}:

$$\ell_{\mathrm{clash}} = \frac{1}{N} \sum_i \left[\max\left(0, r_{\mathrm{clash}} - \min_j \hat{d}_{ij}\right)\right]^2. \tag{36}$$

All peptide–pocket sums and minima are restricted to valid residues. This term provides a coarse interfacial steric regularizer, not an all-atom clash criterion.

Each geometric term is activated when $\tau > \tau_g$ and uses the same time weight as diagonal structural supervision:

$$\mathcal{L}_{\mathrm{aux}} = \mathbb{E}\left[s_{\mathrm{aux}}^2 \sum_{g \in \{\mathrm{bb,pair,int,clash}\}} \mathbf{1}[\tau > \tau_g] \lambda_g w_{\mathrm{time}}(\tau) \ell_g\right]. \tag{37}$$

The scale s_{aux} applies uniformly to auxiliary geometric errors, independently of the weights λ_g. Contact weights are computed from unscaled distances in Å. Geometric loss activation thresholds are specified in Appendix D.4.

A.5 Conditional Packing and Overall Training

Conditioning strategy. The packer in Appendix C.4 predicts side-chain torsions from sequence–backbone conditions and receptor context while keeping the backbone fixed. Following APM (Chen et al., 2025b), we distinguish native-conditioned packing from torsion reconstruction under predicted conditions. Using the diagonal noise time τ, we select

$$Z^{\mathrm{cond}} = \begin{cases} (x_1, R_1, p_1), & \tau < \tau_{\mathrm{pack}}, \\ \mathrm{sg}\left[(\hat{x}_{1;\tau,\tau}, \hat{R}_{1;\tau,\tau}, \hat{p}_{1;\tau,\tau})\right], & \tau \ge \tau_{\mathrm{pack}}, \end{cases} \tag{38}$$

where $p_1 = \mathrm{onehot}(a_1)$. This exposes the packer to diagonal endpoint prediction errors; the switch depends on noise time rather than training stage.

Algorithm 2 Concurrent training of PepMap and its conditional packer

Require: Data $\mathcal{D}$; prior p_0; time laws $q_{\mathrm{diag}}, q_{\mathrm{SG}}$; threshold τ_{pack}; parameters θ, ϕ

repeat
  Sample a minibatch $(C_{\mathrm{rec}}, Z_1, \chi^\star) \sim \mathcal{D}$ and $Z_0 \sim p_0$
  $\tau \sim q_{\mathrm{diag}}, \quad (s,u,t) \sim q_{\mathrm{SG}}$; construct Z_τ, Z_s by Eq. equation 20
  $\widehat{Z}_{1;\tau,\tau} \leftarrow f_\theta(Z_\tau, \tau, \tau, C_{\mathrm{rec}})$
  Compute $\widehat{\mathcal{L}}_{\mathrm{diag}}, \widehat{\mathcal{L}}_{\mathrm{aux}}$ by Eqs. equation 9, equation 37
  $Z_t^{\mathrm{dir}} \leftarrow X^\theta_{s,t}(Z_s)$
  $Z_u^\theta \leftarrow X^\theta_{s,u}(Z_s), \quad \widetilde{Z}_t \leftarrow \mathrm{sg}[X^\theta_{u,t}(Z_u^\theta)]$
  Compute $\widehat{\mathcal{L}}_{\mathrm{SG}}$ with endpoint KL and detached adaptive weights (Appendix A.3)
  $Z^{\mathrm{cond}} \leftarrow \begin{cases} (x_1, R_1, \mathrm{onehot}(a_1)), & \tau < \tau_{\mathrm{pack}}, \\ \mathrm{sg}[\widehat{Z}_{1;\tau,\tau}], & \tau \geq \tau_{\mathrm{pack}} \end{cases}$
  $\hat{\chi} \leftarrow g_\phi(C_{\mathrm{rec}}, \mathrm{sg}[x^{\mathrm{cond}}], \mathrm{sg}[R^{\mathrm{cond}}], \mathrm{sg}[p^{\mathrm{cond}}])$
  Compute $\widehat{\mathcal{L}}_{\mathrm{pack}}$ by Eq. equation 39; FAPE only if $\tau < \tau_{\mathrm{pack}}$
  $\widehat{\mathcal{L}}_{\mathrm{gen}} \leftarrow \widehat{\mathcal{L}}_{\mathrm{diag}} + \lambda_{\mathrm{SG}}\widehat{\mathcal{L}}_{\mathrm{SG}} + \widehat{\mathcal{L}}_{\mathrm{aux}}$
  $\theta \leftarrow \mathrm{OptimizerStep}(\theta, \nabla_\theta \widehat{\mathcal{L}}_{\mathrm{gen}})$
  $\phi \leftarrow \mathrm{OptimizerStep}(\phi, \nabla_\phi \widehat{\mathcal{L}}_{\mathrm{pack}})$
until converged

Ensure: Trained Flow Map X^θ and conditional packer g_ϕ

Packing objective. We use AlphaFold2-style symmetry-aware torsion supervision ℓ_χ and angle-norm regularization ℓ_{norm} (Jumper et al., 2021). The former compares normalized sine–cosine predictions with native torsions, accounting for side-chain symmetry; the latter regularizes the lengths of unnormalized outputs toward one. Both conditioning branches use native torsion labels and valid-angle masks. Under native conditioning, we additionally apply local side-chain FAPE ℓ_{scFAPE} over valid within-residue frame–atom correspondences, using native backbone geometry and residue types with symmetry-aware atom naming. The total objective is

$$\mathcal{L}_{\mathrm{pack}} = \mathbb{E}\Big[\lambda_\chi \ell_\chi + \lambda_{\mathrm{norm}}\ell_{\mathrm{norm}} + \mathbf{1}[\tau < \tau_{\mathrm{pack}}]\,\lambda_{\mathrm{FAPE}}\ell_{\mathrm{scFAPE}}\Big]. \tag{39}$$

Packing loss weights and the conditioning threshold are given in Appendix D.4.

Concurrent optimization. The generator and packer are trained concurrently, with

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}_{\mathrm{diag}} + \lambda_{\mathrm{SG}}\mathcal{L}_{\mathrm{SG}} + \mathcal{L}_{\mathrm{aux}}.$$

Generator losses update θ, including the shared encoder and Pairformer. Packing conditions, frames, and shared representations are detached, so packing losses update only ϕ. Algorithm 2 summarizes training; hatted losses denote masked minibatch estimates, and time sampling and condition selection are performed per example. Receptor conditioning is implicit in X^θ. The packer g_ϕ re-encodes the selected state with the shared encoder and Pairformer, detaches these features, and predicts torsions with fixed frames (Appendix C.4).

A.6 Inference

We initialize $Z_0 = (x_0, R_0, z_0)$ from the priors defined in Appendix A.2, keeping the receptor context fixed throughout generation.

Few-step Flow Map sampling. For $K \in \{1, 5, 10\}$, we apply the learned Flow Map over the uniform time grid $t_k = k/K$ (Algorithm 1):

$$Z_{t_{k+1}} = X^\theta_{t_k, t_{k+1}}(Z_{t_k}), \qquad k = 0, \ldots, K-1.$$

Each transition requires one evaluation of $f_\theta(Z_{t_k}, t_k, t_{k+1}, C_{\mathrm{rec}})$. The final transition directly returns the endpoint prediction, giving K generator evaluations in total. In particular, $K = 1$ produces a direct noise-to-data transition.

Euler sampling. For the 100-NFE setting, we instead integrate the diagonal flow field using Euler updates, with a linear rotational schedule. At time $s < 1$, the network predicts

$$(\hat{x}_{1;s,s}, \hat{R}_{1;s,s}, \hat{p}_{1;s,s}) = f_\theta(Z_s, s, s, C_{\mathrm{rec}}).$$

For a step $0 < h \le 1 - s$, we update

$$\begin{aligned} x_{s+h} &= x_s + \frac{h}{1-s}(\hat{x}_{1;s,s} - x_s), \\ R_{s+h} &= \mathrm{Exp}_{R_s}\left[\frac{h}{1-s}\,\mathrm{Log}_{R_s}(\hat{R}_{1;s,s})\right], \\ z_{s+h} &= z_s + \frac{h}{1-s}(\hat{p}_{1;s,s} - z_s). \end{aligned} \tag{40}$$

The sampler uses 100 generator evaluations in total.

Decoding and accounting. For both samplers, residue identities are decoded after generation from the final endpoint probabilities:

$$p_1^\theta = z_1^\theta, \qquad a_i^\theta = \arg\max_a p_{1,i,a}^\theta.$$

The conditional packer is then applied once, using the generated backbone and endpoint probabilities; the decoded identities determine residue types for atom reconstruction. Packing is excluded from generator NFE but included in wall-clock runtime. Evaluation and timing settings are given in Appendix D.4.

B Properties of the Multimodal Flow Map

We collect several identities needed to interpret PepMap's diagonal and finite-time training objectives. These results specialize standard flow-map properties to PepMap's joint sequence–geometry state and endpoint parameterization. The receptor context C_{rec} and peptide length $N > 0$ are fixed throughout. All predictors condition on the full joint state $Z_t = (x_t, R_t, z_t)$. Network parameters are fixed when considering map composition, and rotational operations are applied residuewise.

B.1 Cross-Entropy Anchoring of the Sequence Drift

Following the CE denoiser interpretation in Discrete Flow Maps (Potaptchik et al., 2026), we specialize the standard conditional-mean characterization to PepMap's joint sequence–geometry state and receptor conditioning. We include the derivation for completeness to clarify why the sequence CE objective anchors the local sequence drift on the diagonal.

Lemma 1 (CE endpoint anchoring under joint conditioning)**.** *Let $Y_i = \mathrm{onehot}(a_{1,i})$, $Y = (Y_i)_{i=1}^N$, and $z_\tau = (1-\tau)z_0 + \tau Y$ follow the training path in Appendix A.2, with $\tau \sim q_{\mathrm{diag}}$. Over all measurable simplex-valued predictors, the expected cross-entropy*

$$\mathcal{R}_{\mathrm{CE}}(p) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N \mathrm{CE}\left(a_{1,i}, p_{\tau,i}(Z_\tau, C_{\mathrm{rec}})\right)\right] \tag{41}$$

is minimized by

$$p^*_{\tau,i}(Z_\tau, C_{\mathrm{rec}}) = \mathbb{E}[Y_i \mid Z_\tau, \tau, C_{\mathrm{rec}}]. \tag{42}$$

Consequently, for $\tau < 1$, the sequence component of the joint marginal drift satisfies

$$\begin{aligned} b^z_\tau(Z_\tau, C_{\mathrm{rec}}) &:= \mathbb{E}[\dot{z}_\tau \mid Z_\tau, \tau, C_{\mathrm{rec}}] \\ &= \frac{p^*_\tau(Z_\tau, C_{\mathrm{rec}}) - z_\tau}{1-\tau}. \end{aligned} \tag{43}$$

Both identities hold almost everywhere under the training law.

Proof. Write $H_\tau = (Z_\tau, \tau, C_{\mathrm{rec}})$ and $q_i = \mathbb{E}[Y_i \mid H_\tau]$. The conditional cross-entropy decomposes as

$$\mathbb{E}[-Y_i^\top \log p_i \mid H_\tau] = H(q_i) + D_{\mathrm{KL}}(q_i \| p_i), \tag{44}$$

where $H(q_i)$ is categorical entropy. Nonnegativity of KL gives the minimum at $p_i = q_i$. Moreover, $\dot{z}_\tau = (Y - z_\tau)/(1-\tau)$, so taking conditional expectations yields Eq. equation 43. No conditional independence between residue identities is required. □

If the endpoint predictor is right-continuous in its target-time argument at $t = s < 1$, the sequence component of the parameterized map in Eq. equation 8 satisfies

$$\partial_t X^{z,\theta}_{s,t}(Z_s)\Big|_{t=s^+} = \frac{\hat{p}_{1;s,s} - z_s}{1-s}. \tag{45}$$

At the population CE optimum, this derivative equals b^z_s almost everywhere under the training law. Thus, the standard CE characterization provides the local justification for PepMap's sequence diagonal objective under joint conditioning. This result concerns the diagonal drift; it does not by itself establish the accuracy or semigroup consistency of the learned finite-time maps. This characterization applies to the unclipped CE objective; the loss clipping used in training is a stabilization heuristic and does not generally preserve the same population optimum.

B.2 Composition Defect of Endpoint-Parameterized Transport

For the sequence component, the following identity is the linear-schedule specialization of the mean-denoiser composition relation in Discrete Flow Maps (Potaptchik et al., 2026). The same affine calculation applies to PepMap's translation component.

Lemma 2 (Affine composition defect)**.** *Fix Z_s and $0 \le s < r < t \le 1$. For an affine component $y \in \{x, z\}$, let*

$$A_y = \hat{y}_{1;s,r}(Z_s), \qquad Z_r = X_{s,r}(Z_s), \qquad B_y = \hat{y}_{1;r,t}(Z_r), \qquad C_y = \hat{y}_{1;s,t}(Z_s),$$

where $\hat{y} = \hat{x}$ for translation and $\hat{y} = \hat{p}$ for sequence. Define

$$\gamma = \frac{(1-\alpha_{r,t})\alpha_{s,r}}{\alpha_{s,t}} = \frac{(1-t)(r-s)}{(1-r)(t-s)} \in [0,1].$$

Then the direct and composed transitions satisfy

$$X^y_{r,t}(X_{s,r}(Z_s)) - X^y_{s,t}(Z_s) = \alpha_{s,t}\left[\gamma A_y + (1-\gamma)B_y - C_y\right]. \tag{46}$$

Consequently, for the fixed state Z_s and time triple (s, r, t), the direct and composed transitions agree in component y if and only if

$$C_y = \gamma A_y + (1-\gamma)B_y.$$

In particular, $A_y = B_y = C_y$ is sufficient but not necessary.

Proof. Let $a = \alpha_{s,r}$, $b = \alpha_{r,t}$, and $c = \alpha_{s,t}$. The time coefficients satisfy

$$(1-b)(1-a) = 1-c, \qquad (1-b)a + b = c.$$

Since $y_r = (1-a)y_s + aA_y$, the composed transition is

$$\begin{aligned} X^y_{r,t}(X_{s,r}(Z_s)) &= (1-b)y_r + bB_y \\ &= (1-c)y_s + (1-b)aA_y + bB_y \\ &= (1-c)y_s + c\left[\gamma A_y + (1-\gamma)B_y\right]. \end{aligned}$$

Subtracting $X^y_{s,t}(Z_s) = (1-c)y_s + cC_y$ gives Eq. equation 46. □

Rotation component. The preceding affine endpoint identity does not directly extend to rotational endpoints on $\mathrm{SO}(3)$. A shared rotational endpoint nevertheless provides a sufficient special case. Suppose that, for every residue, the three calls use the same endpoint $\hat{R}_1$ and

$$\|\operatorname{Log}(R_s^\top \hat{R}_1)\| < \pi.$$

Let

$$a = \alpha_{s,r}, \qquad b = \alpha_{r,t}, \qquad c = \alpha_{s,t} = a + b(1-a),$$

and write

$$v = \mathrm{Log}(R_s^\top \hat{R}_1), \qquad R_r = R_s \,\mathrm{Exp}(av).$$

Then

$$R_r^\top \hat{R}_1 = \mathrm{Exp}((1-a)v), \qquad \mathrm{Log}_{R_r}(\hat{R}_1) = (1-a)v,$$

where the second identity follows from the principal-logarithm assumption. Consequently,

$$\begin{aligned} X^R_{r,t}(X_{s,r}(Z_s)) &= R_r \,\mathrm{Exp}\big(b(1-a)v\big) \\ &= R_s \,\mathrm{Exp}\big([a + b(1-a)]v\big) \\ &= R_s \,\mathrm{Exp}(cv) = X^R_{s,t}(Z_s). \end{aligned}$$

This shared-endpoint condition is sufficient but is neither necessary nor imposed during training. Accordingly, the rotational SG objective compares the direct and composed transported orientations rather than their endpoint predictions.

B.3 Relation to the Interpolation Paths

Consider a fixed-endpoint path with the same affine/geodesic form as Eq. equation 20. For $0 \le s \le t \le 1$, $s < 1$, suppose

$$x_s = (1-s)x_0 + s\hat{x}_1, \qquad z_s = (1-s)z_0 + s\hat{p}_1,$$

and, for each residue,

$$R_s = R_0 \,\mathrm{Exp}(sv_0), \qquad v_0 = \mathrm{Log}(R_0^\top \hat{R}_1), \qquad \|v_0\| < \pi.$$

If $X_{s,t}$ uses this same endpoint $\widehat{Z}_1$, substitution of $\alpha_{s,t}(1-s) = t - s$ gives

$$\begin{aligned} X^x_{s,t}(Z_s) &= (1-t)x_0 + t\hat{x}_1, \\ X^z_{s,t}(Z_s) &= (1-t)z_0 + t\hat{p}_1, \\ X^R_{s,t}(Z_s) &= R_0 \,\mathrm{Exp}\left([s + \alpha_{s,t}(1-s)]v_0\right) = R_0 \,\mathrm{Exp}(tv_0). \end{aligned} \tag{47}$$

Thus the map advances exactly along this fixed-endpoint path. This conditional identity does not require learned marginal-flow trajectories to coincide with individual training interpolants.

B.4 Partition Invariance

The exact composition identity is a defining property of flow maps (Boffi et al., 2024; 2025). Its implication for inference schedules is recorded below.

Corollary 1 (Partition invariance under exact semigroup consistency)**.** *Suppose* $X_{s,s} = \mathrm{Id}$ *and* $X_{r,t} \circ X_{s,r} = X_{s,t}$ *as maps for all* $0 \le s \le r \le t \le 1$. *For a fixed* Z_0*, every finite partition* $0 = t_0 < t_1 < \cdots < t_K = 1$ *satisfies*

$$X_{t_{K-1},1} \circ \cdots \circ X_{t_0,t_1}(Z_0) = X_{0,1}(Z_0).$$

Proof. Starting from $Z_{t_0} = Z_0 = X_{0,0}(Z_0)$, the semigroup identity gives

$$Z_{t_{k+1}} = X_{t_k,t_{k+1}}(X_{0,t_k}(Z_0)) = X_{0,t_{k+1}}(Z_0).$$

Induction yields the claim. □

This result concerns partitions of the same finite-time map, not comparisons with the diagonal-field Euler sampler. Learned maps satisfy semigroup consistency only approximately, so their outputs may depend on the partition.

C Model Architecture

Figures 5 and 6 illustrate the sequence–backbone generator and conditional side-chain packer, respectively. Let $N_{\mathrm{tot}} = N + N_{\mathrm{rec}}$ denote the total number of peptide and receptor residues. We denote residue-level single and pair representations by $\mathbf{h} \in \mathbb{R}^{N_{\mathrm{tot}} \times c_s}$ and $\mathbf{z} \in \mathbb{R}^{N_{\mathrm{tot}} \times N_{\mathrm{tot}} \times c_z}$, respectively, and represent the backbone geometry of residue i by a rigid frame $T_i = (R_i, x_i)$. We use θ and ϕ to denote the parameters of the sequence–backbone generator and packing-specific modules, respectively.

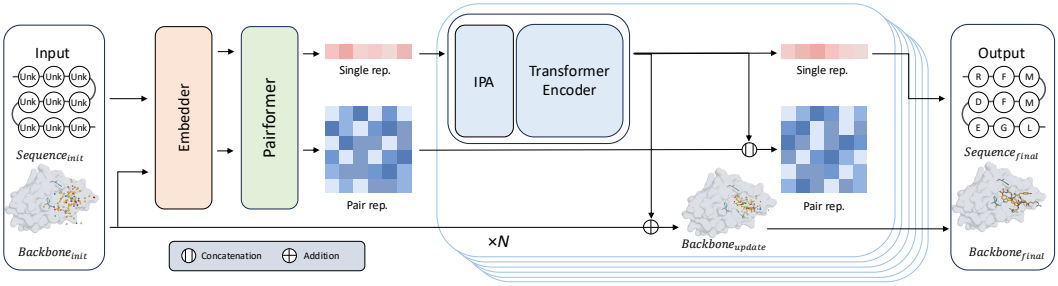


Figure 5: Sequence–backbone generation with a shared feature encoder, Pairformer, and IPA-based endpoint decoder. The encoder receives continuous sequence states; the sequence head outputs categorical endpoint probabilities.

C.1 Feature Encoder

PepMap initializes single representations from residue indices, a binary diffusion indicator, the current continuous peptide sequence state z_s or fixed receptor amino-acid identities, Flow Map time embeddings, and backbone torsion features from the fixed receptor. Pair representations combine projected residue features, relative positional and chain embeddings, pairwise diffusion indicators, C_α–C_α distograms, and torsional context. Peptide torsion features are masked, relative positional features are restricted to same-chain pairs, and distance features are constructed for receptor–receptor, peptide–peptide, and receptor–peptide pairs. The encoding process is summarized as

$$(\mathbf{h}^{\mathrm{enc}}, \mathbf{z}^{\mathrm{enc}}) = \mathrm{Encoder}_\theta(Z_s, C_{\mathrm{rec}};\, s, t-s)\,.$$

The current peptide backbone state (x_s, R_s) and fixed receptor backbone are maintained explicitly as rigid frames alongside the learned representations.

C.2 Pairformer Blocks

MSA-free Pairformer blocks (Abramson et al., 2024) jointly refine the single and pair representations over the full receptor–peptide complex:

$$(\mathbf{h}^{(0)}, \mathbf{z}^{(0)}) = \mathrm{Pairformer}_\theta(\mathbf{h}^{\mathrm{enc}}, \mathbf{z}^{\mathrm{enc}})\,,$$

capturing contextual interactions both within chains and across the receptor–peptide interface.

Table 5: Feature encoder configuration.

Hyperparameter	Value
Single-representation dimension c_s	256
Pair-representation dimension c_z	128
Residue-position embedding dimension	128
Time embedding dimension	128
Amino-acid categories	20
Edge feature dimension	64
Distance bins	22
Maximum relative position	64

Table 6: Pairformer configuration.

Hyperparameter	Value
Blocks	2
Attention heads	8
Multiplicative-update hidden dimension	128
Pair-attention hidden dimension	16
Pair-attention heads	4

C.3 Sequence–Backbone Endpoint Decoder

The refined representations are processed together with the current peptide frames and fixed receptor frames by an IPA-based geometric decoder (Jumper et al., 2021). Let $\mathbf{T}^{(\ell)} = \{T_i^{(\ell)}\}_{i=1}^{N_{\mathrm{tot}}}$ denote the residue-wise rigid frames after ℓ decoder blocks, with $\mathbf{T}^{(0)}$ initialized from the current peptide state and fixed receptor backbone.

For a requested Flow Map interval (s, t), we construct the two-time conditioning

$$\mathbf{c}_{s,t} = [\varphi(s) \,\|\, \varphi(t-s)]\,,$$

where $\varphi(\cdot)$ denotes the time embedding. Each decoder block performs geometry-aware attention followed by time-conditioned residue refinement:

$$\mathbf{u}^{(\ell)} = \mathrm{IPA}_\theta^{(\ell)}\Big(\mathbf{h}^{(\ell)}, \mathbf{z}^{(\ell)}, \mathbf{T}^{(\ell)}\Big)\,,$$
$$\mathbf{h}^{(\ell+1)} = \mathrm{ResUpdate}_\theta^{(\ell)}\Big(\mathbf{h}^{(\ell)}, \mathbf{u}^{(\ell)}; \mathbf{c}_{s,t}\Big)\,,$$

where $\mathrm{ResUpdate}$ comprises the residual IPA update, AdaLN with zero-initialized modulation projections (Peebles & Xie, 2023), sequence Transformer, and node transition.

The updated single representations predict local rigid-frame increments, which are applied only to peptide residues:

$$\Delta T_i^{(\ell)} = \mathrm{BackboneUpdate}_\theta^{(\ell)}\Big(\mathbf{h}_i^{(\ell+1)}\Big)\,,$$
$$T_i^{(\ell+1)} = \begin{cases} T_i^{(\ell)} \circ \Delta T_i^{(\ell)}, & m_i = 1, \\ T_i^{(\ell)}, & m_i = 0, \end{cases}$$

where $m_i = 1$ for peptide residues and $m_i = 0$ for fixed receptor residues. For $T = (R, x)$ and the local increment $\Delta T = (\Delta R, \Delta x)$,

$$(R, x) \circ (\Delta R, \Delta x) = (R\Delta R,\ x + R\Delta x)\,,$$

with Δx expressed in the current residue's local coordinate frame.

Between non-final decoder blocks, pair representations are updated as

$$\mathbf{z}^{(\ell+1)} = \mathrm{EdgeUpdate}_\theta^{(\ell)}\Big(\mathbf{h}^{(\ell+1)}, \mathbf{z}^{(\ell)}\Big)\,, \qquad \ell = 0, \ldots, L-2.$$

After L decoder blocks, the backbone and sequence endpoints are read out as

$$\begin{aligned} \hat{x}_{1;s,t,i} &= x_i^{(L)}, & \hat{R}_{1;s,t,i} &= R_i^{(L)}, \\ \hat{p}_{1;s,t,i} &= \mathrm{Softmax}\Big(\mathrm{SeqHead}_\theta(\mathbf{h}_i^{(L)})\Big)\,, & & i \in \mathcal{I}_{\mathrm{pep}}. \end{aligned} \tag{48}$$

Together, $(\hat{x}_{1;s,t}, \hat{R}_{1;s,t}, \hat{p}_{1;s,t})$ parameterize the finite-time sequence–backbone transport defined in the main text.

Table 7: Decoder and packer IPA configurations.

Hyperparameter	Decoder	Packer
Number of IPA blocks	6	2
IPA hidden dimension	128	128
Number of IPA heads	8	8
Query/key points per IPA head	8	8
Value points per IPA head	12	12
Sequence Transformer attention heads	4	4
Sequence Transformer layers per block	2	2
Backbone rigid update	Yes	No

Table 8: Side-chain torsion prediction head configuration.

Hyperparameter	Value
Hidden dimension	256
Residual blocks	4
Torsion outputs per residue	4
Epsilon	10^{-8}

All configured dropout rates in the Transformer, Pairformer, and IPA modules are set to zero.

C.4 Side-Chain Packing

As illustrated in Figure 6, side-chain packing is applied once after sequence–backbone generation is complete. Let $(x^{\text{gen}}, R^{\text{gen}}, p^{\text{gen}})$ denote the final endpoint output, with $p^{\text{gen}} = z_1^\theta$ the final predicted endpoint probabilities. At inference, the discrete sequence and packing conditioning state are constructed as

$$a_i^{\text{cond}} = \arg\max_a p_{i,a}^{\text{gen}}, \qquad i \in \mathcal{I}_{\text{pep}},$$
$$Z^{\text{cond}} = (x^{\text{gen}}, R^{\text{gen}}, p^{\text{gen}}).$$

The discrete identities specify residue types for final atom reconstruction; the packer receives the predicted endpoint probabilities. Training-time construction of Z^{cond} is described in Appendix A.5.

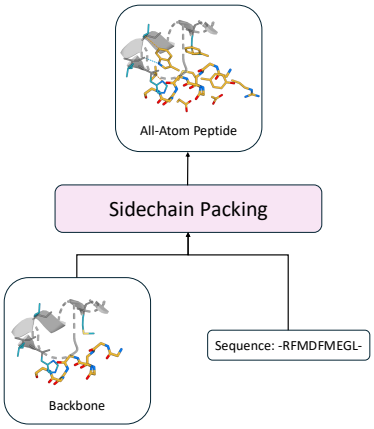


Figure 6: Side-chain packing conditioned on endpoint sequence probabilities, the generated backbone, and receptor context.

The conditioning state is re-embedded together with the fixed receptor using the feature encoder and Pairformer shared with the sequence–backbone generator. With fixed endpoint-time conditioning $s = t = 1$,

$$(\mathbf{h}^{\text{pack}}, \mathbf{z}^{\text{pack}}) = \text{sg}\big[\text{Pairformer}_\theta\big(\text{Encoder}_\theta\big(\text{sg}(Z^{\text{cond}}), C_{\text{rec}};\, 1, 0\big)\big)\big],$$

where $\text{sg}(\cdot)$ denotes stop-gradient. Conditioning inputs and frames are detached, and packing losses therefore update only the packing-specific parameters ϕ.

The detached peptide frames from the selected conditioning state, together with the fixed receptor frames, form the geometric context $\mathbf{T}^{\text{cond}}$. A packing-specific IPA stack refines the representations while keeping these frames fixed:

$$(\mathbf{h}^{\text{pack,out}}, \mathbf{z}^{\text{pack,out}}) = \text{PackIPA}_\phi\big(\mathbf{h}^{\text{pack}}, \mathbf{z}^{\text{pack}}, \mathbf{T}^{\text{cond}}\big).$$

Unlike the sequence–backbone decoder, the packing stack contains no Flow Map time conditioning or rigid-frame updates.

Finally, an AngleResNet-style torsion head (Jumper et al., 2021) predicts up to four side-chain torsions for each peptide residue:

$$\widetilde{\mathbf{a}}_i = \text{TorsionHead}_\phi\Big(\mathbf{h}_i^{\text{pack,out}}, \mathbf{h}_i^{\text{pack}}\Big) \in \mathbb{R}^{4\times 2}. \tag{49}$$

Each row represents a sine–cosine pair for one side-chain torsion. The normalized torsions applicable to each conditioning residue type are combined with the fixed backbone to reconstruct the final full-atom peptide. Training-time torsion supervision and validity masks are detailed in Appendix A.5.

D Experimental Details

D.1 Datasets

We follow the standard preprocessing and splitting protocol of PepBench (Kong et al., 2024). PepBench consists of protein–peptide complexes collected from the PDB, with receptors longer than 30 residues and peptides of 4–25 residues. Complexes are deduplicated at a 90% sequence-identity threshold for both receptor and peptide sequences, resulting in 6,105 non-redundant complexes. We use the expert-curated LNR set (Tsaban et al., 2022) as the test set. After removing complexes containing non-canonical amino acids, 93 test complexes remain. To prevent receptor-level information leakage and evaluate cross-target generalization, receptors are clustered at 40% sequence identity, and all training complexes belonging to receptor clusters shared with the test set are removed. The remaining receptor clusters are then randomly assigned to the training and validation splits, yielding 4,157 training complexes and 114 validation complexes. For each complex, the receptor pocket context C_{rec} is defined as the set of receptor residues whose C_β atoms lie within 10 Å of any native peptide C_β atom. Table 9 summarizes the resulting dataset statistics.

Table 9: PepBench dataset statistics (Kong et al., 2024).

Split	#Complexes	#Receptor clusters	Source
Training	4,157	952	PDB
Validation	114	50	PDB
Test	93	93	LNR (Tsaban et al., 2022)

D.2 Baselines

Unless otherwise stated, peptide lengths match the native peptides. We use official pretrained checkpoints when compatible with PepBench; otherwise, models are retrained on our training split (Appendix D.1) following their official configurations.

RFDiffusion. RFDiffusion (Watson et al., 2023) generates receptor-conditioned peptide backbones, followed by sequence design and all-atom refinement. We use the official pretrained binder-design checkpoint and inference pipeline. For each target, 20% of the binding-site residues are randomly selected as hotspot residues. Generated backbones are subsequently processed by ProteinMPNN (Dauparas et al., 2022) for sequence design and by the official Rosetta refinement pipeline, using one ProteinMPNN design cycle and one Rosetta relaxation cycle.

DiffPepBuilder. DiffPepBuilder (Wang et al., 2024) performs peptide sequence–structure co-design using an SE(3)-equivariant diffusion model. We retrain the model using its default training configuration and disable the disulfide-bond module during generation.

PepFlow. PepFlow (Li et al., 2024) performs full-atom peptide co-design through multimodal flow matching over backbone frames, side-chain torsions, and residue identities. We retrain PepFlow following its default configuration. For co-design, we initialize the peptide at the receptor pocket centroid rather than the native peptide centroid. For side-chain packing, PepFlow is conditioned on the native peptide sequence and backbone while keeping the receptor fixed.

PepGLAD. PepGLAD (Kong et al., 2024) performs full-atom peptide co-design using geometric latent diffusion with a receptor-specific affine transformation. We evaluate the official implementation using the released pretrained checkpoint.

UniMoMo$_{\text{single}}$. We evaluate UniMoMo$_{\text{single}}$ (Kong et al., 2025), the peptide-only variant of the unified block-based latent diffusion framework, using its official implementation and released pretrained checkpoint.

D.3 Evaluation Metrics

For sequence–structure co-design, all methods generate 100 candidates per target, matching the native peptide length. Unless otherwise stated, non-energy metrics are computed on the outputs of each method's generation pipeline, before additional Rosetta relaxation for energy evaluation. Best-candidate selection is performed independently for each metric. Unless otherwise specified for individual metrics, target-level results are averaged with equal weight.

Amino Acid Recovery (AAR). Generated and native sequences are globally aligned using Needleman–Wunsch with BLOSUM62 (Needleman & Wunsch, 1970; Henikoff & Henikoff, 1992), gap-opening and extension penalties of -10.0 and -0.5, and no terminal-gap penalty. For the optimal-scoring alignment, AAR (Luo et al., 2022) is

$$\text{AAR} = \frac{\text{Number of exactly matched amino-acid pairs}}{L_{\text{align}}} \times 100\%, \tag{50}$$

where L_{align} includes gap-containing positions, which count as nonmatches. BLOSUM62 determines the alignment; AAR counts only exact amino-acid matches. We select the highest AAR per target. Higher values indicate better native sequence recovery.

C-RMSD and L-RMSD. For N corresponding peptide C_α atoms, Complex RMSD (C-RMSD) compares generated and native coordinates in the shared receptor frame:

$$\text{C-RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|x_i^{\text{gen}} - x_i^{\text{ref}}\right\|_2^2}. \tag{51}$$

Ligand RMSD (L-RMSD) instead measures error after peptide superposition:

$$\text{L-RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|R_{\text{K}}x_i^{\text{gen}} + t_{\text{K}} - x_i^{\text{ref}}\right\|_2^2}, \tag{52}$$

where $R_{\text{K}} \in \text{SO}(3)$ and $t_{\text{K}} \in \mathbb{R}^3$ are the optimal Kabsch rotation and translation fitted to the corresponding peptide C_α atoms (Kabsch, 1976). C-RMSD therefore reflects both binding-pose displacement and peptide conformational error, whereas L-RMSD assesses conformation after removing global rigid-body differences. Both metrics are in Å, with lower values indicating better recovery. We independently select each target's minimum C-RMSD and L-RMSD and report their cross-target medians.

DockQ. DockQ (Mirabello & Wallner, 2024) combines native-contact recovery and interface and ligand geometry into a score in $[0, 1]$:

$$\text{DockQ} = \frac{1}{3}\left[F_{\text{nat}} + \frac{1}{1+(\text{iRMSD}/1.5\,\text{Å})^2} + \frac{1}{1+(\text{LRMSD}/8.5\,\text{Å})^2}\right]. \tag{53}$$

Here F_{nat} is the fraction of native interchain residue contacts recovered; iRMSD and LRMSD are backbone RMSDs after interface and receptor superposition, respectively. The latter measures ligand placement relative to the receptor, unlike the peptide-superposed L-RMSD above. We use explicit chain mapping and PDB residue-number correspondence while retaining structural superposition, with standard interface settings rather than CAPRI peptide mode. We select the highest DockQ per target without energy filtering. Higher values indicate better docking quality. Target-level scores are averaged with equal weight.

Rosetta Binding Energy (ΔG). We evaluate interface energetics as

$$\Delta G = E_{\text{bound}} - E_{\text{separated}}, \tag{54}$$

where E_{bound} denotes the bound-complex energy and $E_{\text{separated}}$ the combined energy of the separated receptor and peptide after repacking. We obtain `dG_separated` using PyRosetta's `InterfaceAnalyzerMover`. Generated and native complexes first undergo local FastRelax with `ref2015` (Alford et al., 2017), allowing peptide backbone refinement and side-chain optimization in the peptide and its neighborhood while preserving sequence identities. Three Rosetta evaluations are averaged per complex before candidate selection or comparison. We select the lowest energy per target. Scores are in Rosetta Energy Units (REU), with lower values indicating more favorable predicted interface energetics, not experimental binding free energies.

Native-Relative Energy Improvement (IMP). A candidate meets the improvement criterion if

$$\Delta\Delta G = \Delta G_{\text{gen}} - \Delta G_{\text{ref}} \leq 0. \tag{55}$$

where both energies are three-evaluation means obtained using the same relaxation and scoring protocol for the corresponding receptor. IMP is the percentage of evaluated targets with at least one qualifying candidate; equality is included.

Internal and Interface Clashes. We measure C_α crowding with a distance threshold of $3.6574\,\text{Å}$ (Ye et al., 2025). Internal clashes involve another peptide C_α atom, excluding self-pairs and adjacent entries in the extracted coordinate sequence; interface clashes involve any receptor C_α atom. We report

$$\text{Clash}_{\text{in}} = \frac{\text{Number of peptide residues with internal clashes}}{N_{\text{pep}}} \times 100\%, \tag{56}$$

$$\text{Clash}_{\text{out}} = \frac{\text{Number of peptide residues with receptor clashes}}{N_{\text{pep}}} \times 100\%. \tag{57}$$

where N_{pep} counts peptide residues with extracted C_α coordinates. Each residue contributes at most once per metric, regardless of the number of clashes. We average candidate-level rates within each target before cross-target aggregation. Lower values indicate fewer C_α clashes, not necessarily fewer all-atom clashes.

Backbone and Side-Chain Distributional Similarity. JSD_{bb} and JSD_{sc} compare backbone (ϕ, ψ, ω) and side-chain (χ_1–χ_4) angle distributions with UniMoMo's precomputed peptide references (Kong et al., 2025). Valid angles are pooled across all candidates and targets by amino-acid and angle type. Using normalized histograms with thresholds spaced at 10° (Dunbrack & Cohen, 1997), we compute Jensen–Shannon *distance*, the square root of divergence:

$$d_{\text{JS}}(P, Q) = \sqrt{\frac{1}{2} D_{\text{KL}}(P \| M) + \frac{1}{2} D_{\text{KL}}(Q \| M)}, \qquad M = \frac{P + Q}{2}, \tag{58}$$

Here P and Q are generated and reference histograms, and KL terms use natural logarithms. Each metric averages valid residue-type-specific angle distances equally within its torsion group, excluding nonstandard residues and unavailable torsions. This aggregation is over amino-acid–angle types, rather than an equal-weight average of per-target JSD values. Lower values indicate closer agreement of marginal angle distributions.

Sequence and Structural Diversity. For M eligible candidates per target, pairwise distances are

$$d_{\text{seq}}(i, j) = 1 - \frac{I(S_i, S_j) + I(S_j, S_i)}{2}, \tag{59}$$

$$d_{\text{str}}(i, j) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left\| x_i^{(k)} - x_j^{(k)} \right\|_2^2}, \tag{60}$$

where I is fractional sequence identity under the AAR alignment settings, and $x_i^{(k)}$ denotes candidate i's kth corresponding C_α coordinate. Structural distances use N corresponding atoms in the shared receptor frame without additional superposition. Single-linkage hierarchical clustering uses cutoffs of 0.6 for sequence distance and 2 Å for structural distance:

$$\text{Div}_{\text{seq}} = \frac{\text{Number of sequence clusters}}{M}, \tag{61}$$

$$\text{Div}_{\text{str}} = \frac{\text{Number of structural clusters}}{M}. \tag{62}$$

Both metrics use candidates with valid sequences and nonempty C_α coordinates, restricted to the target's most common coordinate length. The denominator M is the number of candidates retained after this filtering, and the resulting cluster proportions are averaged equally across targets. Higher cluster proportions indicate greater diversity.

Undefined values and evaluation failures. Undefined metric values are excluded from their respective aggregates. For DockQ, failures assigned zero scores remain in candidate selection. For diversity, targets with multiple candidates but fewer than two eligible candidates after filtering, or failed clustering, receive zero scores; targets with only one candidate are excluded from diversity aggregation.

D.4 Experimental Settings

Training setup. We implement PepMap in PyTorch and train the generator and conditional packer concurrently from random initialization, without a pretrained teacher. Optimization follows Algorithm 2, with packing losses restricted to the packing-specific parameters as described in Appendix A. Training hyperparameters are summarized in Table 10.

Training configuration. The generator combines diagonal flow matching, semigroup consistency, and auxiliary geometric supervision, with loss weights and geometric settings summarized in Table 11. The packer uses native conditioning for $\tau < \tau_{\text{pack}}$ and detached predicted conditioning otherwise; FAPE is applied only under native conditioning. Weighted FM and adaptively weighted

Table 10: Training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
Optimizer (β_1, β_2)	$(0.95, 0.999)$
Initial learning rate	1×10^{-4}
Minimum learning rate	1×10^{-6}
Weight decay	0.01
Learning-rate schedule	Reduce on plateau
Scheduler factor / patience	0.8 / 10 scheduler checks
EMA decay	0.9999
Batch size per GPU	32
Training GPUs	8 NVIDIA H20-3e GPUs
Training epochs	5000

Table 11: Loss configuration.

Parameter	Value
Loss weights	
FM/SG: translation, rotation, sequence	$(2, 1, 1)$
Overall: FM, SG, auxiliary	$(1, 1, 1)$
Auxiliary: backbone, pairwise, interface, clash	$(1, 1, 1, 5)$
Packing: torsion, norm, FAPE	$(0.5, 0.02, 0.5)$
Geometry	
Coordinate scales (s_x, s_{aux})	$(0.1, 0.1)$
Contact cutoff / transition width	8 Å / 1 Å
Clash threshold	4 Å
Time and weighting	
Time normalization floor δ	0.1
Auxiliary activation τ_g	0.25
Packing switch τ_{pack}	0.8
SG weighting (c, ρ)	$(10^{-3}, 0.5)$
Loss cap	5

SG losses are capped separately for each modality and example, while auxiliary losses are capped after summation, before applying the overall objective weights.

For diagonal supervision, we sample

$$V \sim 0.02\mathcal{U}(0,1) + 0.98\ \mathrm{Beta}(1.9, 1.0), \qquad \tau = 0.01 + 0.99V.$$

For semigroup supervision, we first sample an interval length and then its starting time:

$$\begin{aligned} B &\sim \mathrm{Beta}(1.5, 1.5), & L &= 0.01 + 0.99B, \\ S \mid L &\sim \mathcal{U}(0, 1-L), & T &= S + L, \\ U &= (S+T)/2, & (s, u, t) &= (0.99S,\ 0.99U,\ T). \end{aligned}$$

The final rescaling preserves $s < u < t$, but u is generally not the midpoint of the resulting interval.

Inference setup. We evaluate PepMap at $K \in \{1, 5, 10, 100\}$ generator evaluations. The $K \in \{1, 5, 10\}$ settings use finite-time Flow Map sampling, whereas the 100-NFE setting uses Euler integration with a linear rotational schedule, as specified in Appendix A.6. The final endpoint prediction is included in NFE. Conditional side-chain packing is applied once after generation and is included in wall-clock runtime but excluded from generator NFE. Candidate counts and peptide-length conditions follow the evaluation protocol in Appendix D.3. For the corresponding comparisons in Figure 3, PepGLAD and UniMoMo$_{\text{single}}$ are evaluated using DDIM with $\eta = 0$ at 1, 5, and 10 steps, and DDPM at 100 steps.

Runtime measurement. For Figure 3, runtime is measured on a single NVIDIA H20-3e GPU with batch size 1 and averaged over all 93 LNR targets. Timing includes sampling and native full-atom reconstruction, but excludes checkpoint loading, input feature loading, and output file writing. Each configuration uses two untimed warm-up generations, with CUDA synchronization before and after each timed call.

E Additional Experiments

E.1 Few-Step Generation Comparisons

We compare PepMap with DiffPepBuilder, PepFlow, PepGLAD, and UniMoMo$_{\text{single}}$ at matched budgets of one and ten generator evaluations (Tables 12 and 13). PepGLAD and UniMoMo$_{\text{single}}$ use DDIM for low-NFE sampling; standard-budget baseline results are provided in Tables 1 and 2. Evaluation follows Appendix D.3, with RMSDs in Å and clash rates reported as percentages of C_α crowding rather than all-atom clashes. At both budgets, PepMap achieves the lowest C-RMSD, L-RMSD, and internal clash rate, together with the highest DockQ. Notably, one-step PepMap outperforms all four baselines at ten evaluations on C-RMSD, L-RMSD, and DockQ, demonstrating its effectiveness under restricted inference budgets. Although PepMap's interface clash rate decreases from 9.41% to 6.16%, it remains higher than that of PepFlow at one evaluation and UniMoMo$_{\text{single}}$ at ten, indicating room for improved receptor–peptide steric compatibility.

Table 12: LNR few-step generation at 1 NFE.

Model	C-RMSD↓	L-RMSD↓	DockQ↑	Clash_{in}↓	$\text{Clash}_{\text{out}}$↓
DiffPepBuilder	4.12	2.03	0.615	48.54%	15.80%
PepFlow	4.48	1.95	0.551	31.92%	**7.85%**
PepGLAD	11.16	4.36	0.248	16.04%	22.91%
UniMoMo$_{\text{single}}$	19.10	12.32	0.132	6.80%	17.34%
PepMap	**2.52**	**1.12**	**0.735**	**4.12%**	9.41%

Table 13: LNR few-step generation at 10 NFE.

Model	C-RMSD↓	L-RMSD↓	DockQ↑	Clash_{in}↓	$\text{Clash}_{\text{out}}$↓
DiffPepBuilder	4.76	1.77	0.550	4.87%	12.44%
PepFlow	4.16	1.66	0.561	2.13%	7.05%
PepGLAD	3.40	1.73	0.641	22.30%	8.52%
UniMoMo$_{\text{single}}$	2.87	1.88	0.712	2.56%	**4.85%**
PepMap	**2.43**	**1.05**	**0.762**	**0.62%**	6.16%

E.2 Side-Chain Packing Comparisons

We compare PepMap's packer with Rosetta under native sequence, backbone, and receptor conditioning (Table 14). We also evaluate both packers on the same sequences and backbones generated by PepMap at 100 NFE (Table 15). Under native conditioning, Rosetta has lower MAE for χ_1–χ_4 and higher Angle Accuracy (48.73% vs. 16.62%), indicating more accurate native-rotamer recovery. On PepMap-generated inputs, PepMap yields a lower JSD_{sc} (0.285 vs. 0.438), whereas Rosetta yields a lower post-relaxation ΔG (-31.09 vs. -27.56 REU) and higher IMP (32.66% vs. 28.18%). Because FastRelax reoptimizes the peptide backbone and nearby side chains, the energy comparison does not isolate the initial packing step. These results distinguish native-rotamer recovery and distributional similarity from post-relaxation interface energetics.

Table 14: Packing under native conditioning.

Method	Angle MAE (°) ↓				Angle Accuracy (%) ↑
	χ_1	χ_2	χ_3	χ_4	
PepMap	58.17	46.89	65.57	69.52	16.62
Rosetta	31.78	30.99	48.27	57.84	48.73

Table 15: Packing on PepMap outputs.

Packer	JSD_{sc} ↓	ΔG ↓	IMP (%) ↑
PepMap packer	0.285	-27.56	28.18%
Rosetta	0.438	-31.09	32.66%

E.3 Extended Ablation Analysis

All variants retain diagonal flow-matching supervision and use the same training budgets and evaluation protocols. We separately disable semigroup supervision ($\lambda_{\text{SG}} = 0$), remove all four auxiliary geometric losses ($\mathcal{L}_{\text{aux}} = 0$), or bypass Pairformer while retaining receptor conditioning, the IPA trunk, and prediction heads. Each configuration is trained once. Without semigroup supervision, increasing K from 1 to 10 improves C-RMSD only from 5.21 to 5.01 Å and DockQ from 0.513 to

0.534, remaining worse than full-model single-step results. Although internal clashes and backbone JSD improve with more steps, additional sampling does not compensate for the missing supervision within the tested budgets. Removing Pairformer worsens all five metrics at both budgets, supporting its contribution to few-step co-design, although this ablation does not separate representation refinement from model-capacity effects. Removing geometric supervision increases interface clashes from 9.41%/6.16% to 16.25%/14.11% at $K = 1/10$, alongside worse C-RMSD and DockQ, while backbone JSD changes only slightly. Internal clashes decrease at $K = 1$ (4.12% to 3.71%) but increase at $K = 10$ (0.62% to 1.57%). The combined geometric objective therefore benefits pose recovery and interface compatibility without uniformly improving internal geometric metrics.

F Additional Visualizations

Figure 7 presents PepMap-generated peptide structures across sampling budgets for five receptor targets, alongside their reference complexes.

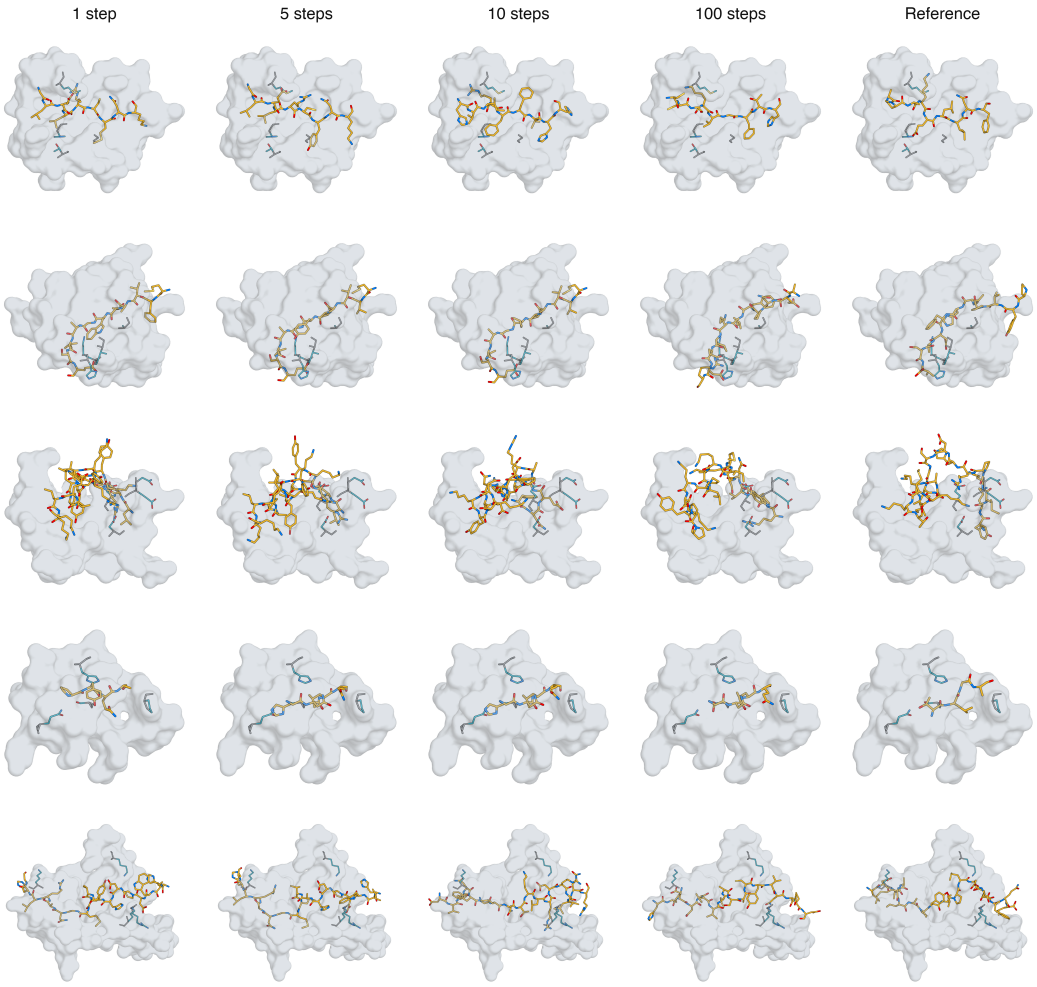


Figure 7: PepMap generation across sampling budgets. Rows show 2WV5, 1SSC, 3R7G, 4FVD, and 5L9B (top to bottom); columns show outputs at 1, 5, 10, and 100 NFE, followed by the reference complexes. Structures within each row are receptor-aligned and shown from the same viewpoint, with light-gray receptor surfaces and yellow peptide carbon atoms. Sampling uses Flow Maps at 1–10 NFE and Euler integration at 100 NFE.

G LIMITATIONS AND FUTURE WORK

Semigroup consistency does not guarantee molecular validity, and PepMap remains limited by receptor–peptide steric compatibility. Its factorized pipeline packs side chains on a fixed backbone without feedback to generation. Fine-tuning packing models on receptor–peptide complexes could improve reconstruction and interface geometry; effects on predicted energetics require controlled evaluation. Alternatively, extending pocket-conditioned finite-time transport to VAE-based all-atom representations, inspired by PepGLAD, UniMoMo, and La-Proteina (Kong et al., 2024; 2025; Geffner et al., 2025), could integrate learned reconstruction priors into few-step generation. PepMap also requires a specified pocket and peptide length; adapting variable-length generation (Cheng et al., 2026) to few-step transport could broaden its applicability. Inference-time reward guidance remains unexplored: flow-map guidance or reward-based resampling (Huang et al., 2026; Hartman et al., 2025) could target interface geometry and predicted energetics, with trade-offs among reward improvement, sample diversity, and computational cost. Finally, structural and energy metrics do not establish binding affinity or specificity, which require broader target evaluation and experimental validation.