PEPFM-TANRL: FINE-TUNING A MULTI-MANIFOLD FLOW MATCHING MODEL FOR PEPTIDE DESIGN VIA TANGENT-SPACE GRPO

Anonymous authors
Paper under double-blind review

ABSTRACT

De novo design of peptides that bind a target protein is a discrete–continuous generative problem requiring the joint modeling of structure and sequence. Flow-matching-based generative models faithfully reproduce the distribution of native peptide–receptor complexes, yet likelihood training provides no guarantee that generated samples exhibit high binding affinity. We introduce PepFM-TanRL, a framework that brings Group Relative Policy Optimization (GRPO) to deterministic-ODE flow models operating over multi-modal continuous state spaces. Injecting time-decaying noise converts the ODE sampler into a stochastic policy, from which we derive factored importance ratios and analytic KL penalties across four modalities—$\mathbb{R}^3$, $\mathrm{SO}(3)$, torus, and logit space. As an architectural change that enables reinforcement learning, we predict torsion angles for all 20 amino acid types simultaneously and aggregate side-chain features via probability-weighted expectations, eliminating discrete sampling so that every generation step admits a closed-form Gaussian log-likelihood in the tangent space. Fine-tuning with a reward that splices the Vinardo docking score with a steric-clash penalty improves the reward distribution on all four targets (best-target median reward $-23.9 \rightarrow -0.4$; clash-free fraction $39\% \rightarrow 80\%$); on the one target that starts with a very high clash rate, the gain stops at fewer clashes without reaching a Vinardo-scorable population. On the other three, Vinardo scores among clash-free samples also shift higher as energetically unfavorable conformations are suppressed, while native-like backbone geometry and structural diversity are preserved. We further analyze how reinforcement learning alters the similarity of generated structures to the native complex.

1 INTRODUCTION

Peptides occupy a middle ground between small molecules and antibodies as a therapeutic modality, combining high specificity from extensive contact surfaces with relatively low manufacturing cost (Muttenthaler et al., 2021). Given the binding pocket of a target protein, simultaneously generating a peptide sequence and its bound-state structure is formulated as a *conditional co-design* problem (Li et al., 2024; Kong et al., 2024).

Recent approaches based on diffusion models and flow matching treat the backbone rigid-body transforms (translation and rotation), side-chain torsion angles, and amino acid sequence as generative processes on their respective manifolds, faithfully reproducing the distribution of experimentally determined peptide–receptor complexes (Watson et al., 2023; Li et al., 2024). These models are trained to maximize the likelihood of the training data, so their generative distributions are optimized for "native-likeness" rather than for binding affinity itself. The metrics that certify such training inherit the same limitation: recovering the single crystallized sequence more closely does not imply stronger binding. In our measurements, more than half of the samples from the pretrained model contain steric clashes (heavy-atom distances < 1.1 Å), with the clash-free fraction ranging from 3% to 48% across four targets, causing docking scores to be unevaluable and the median reward to saturate at its lower bound (Table 2).

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.

This gap mirrors the relationship between pretraining and preference alignment in large language models. The standard remedy is to optimize non-differentiable rewards directly via reinforcement learning (Ouyang et al., 2022), and the same strategy has proven effective for fine-tuning diffusion models in image generation (Black et al., 2024; Fan et al., 2023). Group Relative Policy Optimization (GRPO) is particularly suitable: it requires no value function and estimates the advantage from the reward statistics of multiple samples drawn under the same condition (Shao et al., 2024). In peptide design, rewards such as docking scores and clash counts are physics-based score functions—non-differentiable yet cheap to evaluate per sample—a setting that aligns well with GRPO's assumptions.

However, applying this idea to a flow-matching model for all-atom structure generation requires solving three problems that do not arise in language models. First, when the generative process is defined as a deterministic ODE, the policy probability degenerates and importance ratios are undefined. Second, the state lives on a product of heterogeneous manifolds—$\mathbb{R}^3$, $\mathrm{SO}(3)$, torus, and the sequence logit space—so naïvely treating the joint log-likelihood as a single scalar creates extreme imbalances in dimensionality and scale across modalities. Third, samples for which the reward cannot be evaluated (physically meaningless structures with self-intersections) inevitably arise during generation, leaving the reward function undefined over part of its domain. We resolve these issues in our implementation PepFM-TanRL and experimentally verify that physics-based rewards lead to improved reward distributions.

On all four targets, both the training reward and physical validity improve clearly. As independent validation, re-evaluating generated structures with a score function not used during training—the unrelaxed Rosetta binding free energy ΔG_{bind} (Alford et al., 2017)—confirms improved binding affinity on every target, consistent with the direction of Vinardo reward improvement.

Our contributions are threefold. First, we construct a practical framework for applying GRPO to flow models with deterministic ODEs and multi-modal continuous state spaces (Section 4). We reinterpret the flow model as a stochastic policy by injecting time-dependent noise, derive per-modality importance ratios from isotropic Gaussian transition kernels on each manifold's tangent space, obtain analytic KL penalties to the reference policy, and splice an exponentially decaying clash penalty onto the docking score to continuously fill the region where the reward is otherwise undefined. Second, we introduce an architectural change that enables reinforcement learning: predicting torsion angles for all 20 amino acid types simultaneously and aggregating side-chain features via probability-weighted expectations, eliminating discrete sampling so that every generation step admits a closed-form Gaussian log-likelihood for importance-ratio computation (Section 3). Third, we demonstrate improved reward distributions on all four targets under this framework and confirm the improvement with an independent score function (Section 5).

2 Related Work

Generative design of peptides and proteins. Diffusion-based protein design has reached practical maturity through RFdiffusion (Watson et al., 2023) and sequence-space diffusion (Lisanza et al., 2025), with $\mathrm{SE}(3)$ diffusion and flow models (Yim et al., 2023; Bose et al., 2024) becoming the standard framework for backbone generation. The challenge specific to peptides is that backbone geometry alone does not determine the binding mode of short chains; sequence and side chains must be modeled jointly rather than in the two stages of backbone generation followed by inverse folding (Dauparas et al., 2022). MultiFlow (Campbell et al., 2024) couples a discrete flow over sequence with a continuous flow over backbone frames for protein co-design; PepGLAD (Kong et al., 2024) addresses the peptide setting via geometric latent diffusion, while PepFlow (Li et al., 2024) formulates full-atom co-design as Riemannian flow matching (Chen & Lipman, 2024; Lipman et al., 2023) over $\mathbb{R}^3$, $\mathrm{SO}(3)$, torus, and the probability simplex. We build on the latter, but all of these methods aim to reproduce the data distribution and lack a mechanism for incorporating physics-based objectives as learning signal. Our contribution is to provide a framework for post-hoc optimization of the pretrained model.

Reward-based fine-tuning of generative models. DDPO (Black et al., 2024) and DPOK (Fan et al., 2023) cast the reverse process of diffusion models (Ho et al., 2020) as a multi-step MDP, enabling policy gradients with non-differentiable rewards. When the reward is differentiable, back-propagating through it directly (Clark et al., 2024) is more efficient, but docking scores are external

programs and preclude this. The same policy-gradient view has been carried over to biological design, where the bottleneck is the cost of each reward query rather than its differentiability (Uehara et al., 2024). Extensions to flow matching include Adjoint Matching (Domingo-Enrich et al., 2025), which frames fine-tuning as stochastic optimal control, and Flow-GRPO (Liu et al., 2025a), which converts the ODE to an SDE and applies GRPO. Our stochastic-policy construction shares the same intuition as Flow-GRPO, but the state space is a product of distinct manifolds rather than a single Euclidean space, necessitating factored importance ratios across modalities. GRPO itself was proposed in the context of language models (Shao et al., 2024) and uses the PPO clipping objective (Schulman et al., 2017) without a value function.

Physics-based rewards as imperfect proxies. The poor absolute predictive power of docking scores for measured binding affinities has long been noted (Warren et al., 2006), and even improved scoring functions such as Vinardo (Quiroga & Villarreal, 2016) have limited ranking ability. In small-molecule generative design, optimizing such a score without constraint is known to yield compounds that score well while being chemically implausible (Renz et al., 2019; Ciepliński et al., 2023); the analogous over-optimization of *learned* reward models has been studied quantitatively (Gao et al., 2023; Skalse et al., 2022). We therefore do not treat the docking score as an objective to be maximized without bound. Because we fine-tune rather than train from scratch, the pretrained flow model already concentrates its mass on native-like peptide structures, and the per-modality KL penalty toward the frozen pretrained policy (Section 4) keeps the fine-tuned policy anchored to that distribution. The reward is thus used to improve the reward statistics of the generated ensemble *within* the range of structures the pretrained model regards as natural, rather than to search for the optimum of the score function itself.

3 PRELIMINARIES

3.1 PROBLEM SETUP AND MULTI-MODAL FLOW MATCHING

Given a receptor binding pocket $\mathcal{C}^{\mathrm{rec}}$, we represent an n-residue peptide as $\mathcal{C}^{\mathrm{pep}} = \{(a^j, R^j, x^j, \chi^j)\}_{j=1}^n$, where $x^j \in \mathbb{R}^3$ is the C_α coordinate of residue j, $R^j \in \mathrm{SO}(3)$ is the backbone frame orientation, $\chi^j \in \mathbb{T}^5$ contains up to five side-chain torsion angles on the torus $\mathbb{T} = [-\pi, \pi)$, and a^j is the amino acid type. The goal is to sample from the conditional distribution $p(\mathcal{C}^{\mathrm{pep}} \mid \mathcal{C}^{\mathrm{rec}})$, with an emphasis on obtaining samples that exhibit high binding affinity and physical plausibility.

The pretrained model learns each modality via conditional flow matching (Lipman et al., 2023; Chen & Lipman, 2024) on its respective manifold. For $\mathbb{R}^3$ and $\mathrm{SO}(3)$, the velocity field follows geodesic interpolation ψ_t; for $\mathbb{T}$, it uses the shortest-path velocity with $\texttt{wrap}(u) = (u + \pi) \bmod 2\pi - \pi$; and for the sequence, linear interpolation in a logit space corresponding to Δ^{19} is employed. The network f_θ receives the state $\mathcal{C}_t^{\mathrm{pep}}$ at time t together with $\mathcal{C}^{\mathrm{rec}}$ and predicts the terminal state $\hat{\mathcal{C}^{\mathrm{pep}}}_1$. At generation time, the ODE is integrated from $t = 0$ to $t = 1$ using Euler steps of size $1/N$.

To enable reinforcement learning, we introduce the following architectural modifications relative to PepFlow and retrain from scratch (see Appendix F for full details).

Joint prediction of torsion angles for all amino acid types. The original PepFlow represents torsion angles as a single five-element vector shared across amino acid types and, at each step, discretely samples one type from the predicted logits to interpret the torsion values. Because torsion angle distributions differ across amino acid types, angles predicted for one type (e.g., proline) may cause steric clashes when interpreted as another (e.g., lysine). Moreover, the discrete sampling operation breaks the Gaussian structure of the tangent-space transition kernel, preventing the closed-form log-likelihood computation required for importance-ratio-based reinforcement learning. Our model instead predicts and maintains 20×5 torsion angles per residue—one set for each amino acid type—simultaneously. Features are constructed as a probability-weighted expectation over types, using the predicted amino acid probabilities $p_t(a) = \mathrm{softmax}(\hat{s}_{1,\theta})$:

$$h_j^{\mathrm{sc}} = \sum_{a=1}^{20} p_t(a^j)\, h_{j,a}^{\mathrm{sc}} \tag{1}$$

where $h^{\text{sc}}_{j,a}$ is the feature vector computed from the torsion angles of amino acid type a (Appendix F.2). This design (1) structurally eliminates cross-type confusion of angle distributions, and (2) maintains a connected gradient graph from the state at time t to the prediction at $t + \Delta t$, so that the only stochastic element in the generation process is the injected noise described below.

The sequence loss uses the velocity-field MSE parameterization (`MSE`) described in PepFlow; we also investigate a cross-entropy variant (`CE`) and compare the two in Appendix B. The structural losses are identical to PepFlow: translation, rotation, and backbone atom coordinate terms with weights $0.5, 0.5, 0.25$; torsion velocity field at 1.0 with its auxiliary term (distance to the predicted $t = 1$ angle) at 0.5; and the sequence term at 1.0.

3.2 Group Relative Policy Optimization

GRPO (Shao et al., 2024) generates G samples $\{\mathcal{C}^{\text{pep},(i)}\}_{i=1}^{G}$ from the same condition and normalizes the advantage using within-group reward statistics:

$$A^{(i)} = \frac{r^{(i)} - \text{mean}(\{r^{(j)}\})}{\text{std}(\{r^{(j)}\}) + \varepsilon_{\text{num}}}, \qquad \varepsilon_{\text{num}} = 10^{-6} \tag{2}$$

Since no value function is learned, GRPO is particularly well-suited to settings where the reward is an external program call. For targets where the reward distribution concentrates at its lower bound and the within-group standard deviation becomes small, the denominator in equation 2 erases the distinction between moderately and substantially better samples. In such cases we use a variant without standard-deviation normalization: $A^{(i)} = r^{(i)} - \text{mean}(\{r^{(j)}\})$ (Section 5.1), the same modification that Liu et al. (2025b) adopt to remove the difficulty bias that std-normalization introduces in language-model GRPO.

4 Method

4.1 Reinterpreting the Flow Model as a Stochastic Policy

A deterministic sampler uniquely determines the trajectory from the initial noise, causing the policy density to degenerate and rendering importance ratios undefined. We resolve this by injecting time-dependent noise at each Euler step. For modality $m \in \{\text{pos}, \text{ori}, \text{ang}, \text{type}\}$, the transition from time t is defined as

$$\mathcal{C}^{\text{pep},m}_{t+\Delta t} \;=\; \texttt{exp}_{\mu^m_\theta}\big(\sigma_m(t)\sqrt{\Delta t}\,\epsilon\big), \qquad \epsilon \sim \mathcal{N}(0, I_{d_m}) \tag{3}$$

where `exp` denotes the exponential map on the respective manifold (addition for $\mathbb{R}^3$ and logit space; the exponential maps of $\text{SO}(3)$ and $\mathbb{T}$ otherwise), and μ^m_θ is the deterministic next state obtained by geodesic interpolation from $\mathcal{C}^{\text{pep},m}_t$ toward the model's predicted terminal state $\hat{\mathcal{C}^{\text{pep},m}}_{1,\theta}$:

$$\mu^m_\theta \;=\; \texttt{exp}_{\mathcal{C}^{\text{pep},m}_t}\left(\frac{\Delta t}{1-t}\,\texttt{log}_{\mathcal{C}^{\text{pep},m}_t}\big(\hat{\mathcal{C}^{\text{pep},m}}_{1,\theta}\big)\right) \tag{4}$$

The noise magnitude decays linearly as $\sigma_m(t) = \sigma^{\min}_m + (\sigma^{\max}_m - \sigma^{\min}_m)(1-t)$ with $\sigma^{\min}_m = 0$, so that the stochastic sampler reduces to the deterministic one as $t \to 1$. This construction mirrors an Euler–Maruyama discretization of the generative ODE augmented with a Wiener process. Because the state space includes $\text{SO}(3)$ and $\mathbb{T}$, a standard Itô SDE is not available; we instead project isotropic Gaussian noise from the tangent space onto the manifold via the exponential map. The one-step transition kernel is thus an isotropic Gaussian in each manifold's tangent space, yielding a closed-form log-likelihood. To promote diversity, we perturb each group's C_α centroid (std 2.0 Å) and residue count (± 2).

4.2 Factored Importance Ratios

Treating the joint likelihood as a single scalar causes numerical instability from dimension and scale imbalances across modalities, so we factor the formulation per modality. For modality m and residue j, the normalized noise of the realized transition $\mathcal{C}^{\text{pep}}_t \to \mathcal{C}^{\text{pep}}_{t+\Delta t}$ is

$$\xi^m_{\theta,j} \;=\; \frac{\texttt{log}_{\mu^m_\theta}\big(\mathcal{C}^{\text{pep},m}_{t+\Delta t}\big)_j}{\sigma_m(t)\sqrt{\Delta t}} \tag{5}$$

Algorithm 1 Stochastic generation from noise. $\sigma_m(t) = \sigma_m^{\max}(1-t)$ decays to zero at t=1. Setting $\sigma_m^{\max} = 0$ recovers deterministic generation.

Require: Receptor $\mathcal{C}^{\text{rec}}$; model f_θ; number of steps N
Ensure: Generated peptide $(a,\ R,\ x,\ \chi)$

$h_0 \leftarrow \textsc{NodeEmbed}(\mathcal{C}^{\text{rec}})$; $z_0 \leftarrow \textsc{EdgeEmbed}(\mathcal{C}^{\text{rec}})$ ▷ once; peptide slots blank
Sample $\mathcal{C}_0^{\text{pep}}$: $x \sim \mathcal{N}(0, I)$, $R \sim \text{Unif}(\text{SO}(3))$, $\chi \sim \text{Unif}(\mathbb{T}^{20\times 5})$, $s \sim \mathcal{N}(0, I)$
$\Delta t \leftarrow 1/N$
for $n = 0, \ldots, N-1$ **do**
  $t \leftarrow n \cdot \Delta t$
  $(\hat{s}_1, \hat{R}_1, \hat{x}_1, \hat{\chi}_1) \leftarrow f_\theta(\mathcal{C}_t^{\text{pep}},\ t,\ h_0,\ z_0)$ ▷ Alg. 3
  for each modality m **do**
    $\mu_\theta^m \leftarrow$ geodesic interpolation equation 4
    $\epsilon \sim \mathcal{N}(0, I_{d_m})$; $\mathcal{C}_{t+\Delta t}^{\text{pep},m} \leftarrow \texttt{exp}_{\mu_\theta^m}\big(\sigma_m(t)\sqrt{\Delta t}\ \epsilon\big)$ ▷ equation 3
  end for
end for
return $\big(\text{argmax}(\hat{s}_1),\ R,\ x,\ \hat{\chi}_1\big)$

where $\texttt{log}$ is the logarithmic map on the corresponding manifold. The importance ratio is then

$$\rho_j^m \ = \ \exp\Big(\tfrac{1}{2}\big(\|\xi_{\text{old},j}^m\|^2 - \|\xi_{\theta,j}^m\|^2\big)\Big) \tag{6}$$

The Gaussian normalization constant, which depends on $\sigma_m(t)$, cancels in the ratio (Appendix A). Applying PPO-style clipping (Schulman et al., 2017), we define the per-modality surrogate objective as

$$\mathcal{L}^m = \mathbb{E}\Big[\ \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} \min\big(\rho_j^m A,\ \text{clip}(\rho_j^m, 1-\epsilon, 1+\epsilon)A\big)\Big] \tag{7}$$

where $\mathcal{J}$ is the set of generated residues, and averaging over $|\mathcal{J}|$ normalizes gradient scale across peptides of different lengths.

Torsion angle handling. Because the relevant angles depend on the amino acid type, the torsion importance ratio replaces the exponent in equation 6 with $\frac{1}{2}\sum_a \sum_k p_t(a)\,\mathbb{1}[a,k]\big(\|\xi_{\text{old}}\|^2 - \|\xi_\theta\|^2\big)_{a,k}$, weighting the 20×5 log-likelihood differences by $p_t(a)$ and a validity mask $\mathbb{1}[\chi_k \text{ exists for } a]$.

4.3 KL Regularization toward the Reference Policy

To keep the policy anchored to the native-like distribution learned during pretraining—and thereby prevent the reward from being gamed outside it—we impose a per-modality KL penalty toward the frozen pretrained reference policy π_{ref}. Since both transition kernels share the same isotropic Gaussian variance, the KL reduces to a squared distance between means:

$$\text{KL}^m = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} \frac{\big\|\texttt{log}_{\mu_\theta^m}\big(\mu_{\text{ref}}^m\big)_j\big\|^2}{2\,\sigma_m(t)^2\,\Delta t} \tag{8}$$

The overall optimization objective combines the per-modality surrogates and KL terms with modality weights λ^m and KL coefficient β:

$$\mathcal{L}(\theta) = -\sum_m \lambda^m\big(\mathcal{L}^m - \beta\,\text{KL}^m\big) \tag{9}$$

For targets with extremely sparse rewards, we down-weight groups whose best sample still falls below r_0 via $w_g = \sigma\big((\max_{i\in g} r^{(i)} - r_0)/\tau_w\big)$ (σ: sigmoid, r_0=−20, τ_w=2.0), applied to both $\mathcal{L}^m$ and KL^m, so that learning effectively proceeds only from groups containing at least one clash-free sample.

Algorithm 2 GRPO fine-tuning of the stochastic flow model.

Require: Pretrained θ_0; frozen reference $\theta_{\text{ref}} \leftarrow \theta_0$; reward $r(\cdot)$ equation 10
Ensure: Fine-tuned θ

$\theta \leftarrow \theta_0$
repeat
  $\theta_{\text{old}} \leftarrow \theta$ ▷ sync weights to sampler
  for $g = 1, \ldots, B$ **do** ▷ fill replay buffer
    Generate G samples via Alg. 1 with θ_{old}; store trajectories and per-step ξ^m_{old}
    Evaluate $\{r^{(i)}\}$ in parallel; compute $A^{(i)}$ equation 2
  end for
  for epoch $= 1, \ldots, E$ **do** ▷ policy update
    Sample batch of transitions $(t,\, \mathcal{C}^{\text{pep}}_t \to \mathcal{C}^{\text{pep}}_{t+\Delta t},\, A)$ from buffer
    for each modality m **do**
      Compute ρ^m_j from ξ^m_{old} and ξ^m_θ equation 6; $\mathcal{L}^m \leftarrow$ clipped surrogate equation 7; KL^m equation 8
    end for
    $\mathcal{L}(\theta) \leftarrow -\sum_m \lambda^m \big(\mathcal{L}^m - \beta\, \mathrm{KL}^m\big)$ ▷ equation 9
    $\theta \leftarrow \theta - \alpha\, \nabla_\theta \mathcal{L}(\theta)$
  end for
until convergence

4.4 Physics-Based Reward

The reward splices a docking score term with a steric clash penalty, switching between the two based on whether clashes are present. From heavy-atom distances (excluding hydrogens), we count intra-peptide clashes n_{intra} and peptide–receptor clashes n_{inter} at threshold d_{th}. The Vinardo score S_{vd} (Quiroga & Villarreal, 2016), computed with the AutoDock Vina 1.2 Python bindings (Eberhardt et al., 2021), is evaluated only when no clashes are present; otherwise a penalty is assigned:

$$r(\mathcal{C}^{\text{pep}}) = \begin{cases} \max\big(-S_{\text{vd}}(\mathcal{C}^{\text{pep}}),\; r_{\min}\big) & n_{\text{intra}} = n_{\text{inter}} = 0 \\ r_0 + (r_{\text{worst}} - r_0)\big(1 - e^{-\kappa_1 n_{\text{intra}} - \kappa_2 n_{\text{inter}}}\big) & \text{otherwise} \end{cases} \tag{10}$$

$r_0 = r_{\min} = -20$ is the baseline at zero clashes and $r_{\text{worst}} = -30$ is the penalty floor. This ensures a finite, monotone reward with a gradient toward fewer clashes everywhere. Before computing S_{vd}, hydrogens are added and a brief energy minimization is performed with heavy atoms restrained (PDBFixer / OpenMM (Eastman et al., 2017)). Samples with ring structures inconsistent with the generated sequence are discarded.

In the later phase of training, concentration of the reward distribution at its lower bound stalls learning. We address this with exponential reward rescaling combined with unnormalized advantage (Appendix C).

4.5 Training Procedure

Each iteration synchronizes the current policy weights to a sampler that generates groups of $G = 32$ samples and pushes them to an asynchronous replay buffer. Reward evaluation runs in parallel, overlapping with sampling. The buffer stores up to eight groups; equation 9 is updated on buffered groups for up to 64 iterations, amortizing the cost of each docking call over many gradient steps. Algorithm 2 summarizes the training loop; key hyperparameters are listed in Appendix Table 4. Further details on the distributed training infrastructure and per-target configuration are in Appendix D.

5 Experiments

5.1 Experimental Setup

Pretraining. We pretrain on the preprocessed dataset released by PepFlow (Li et al., 2024) for 40,000 steps under each sequence parameterization (Adam, lr 5×10^{-4}, batch 32, 8 GPUs). On a

Table 1: Targets and training configurations. Pocket length is the number of receptor-side input tokens. "Switch" denotes the step at which exponential reward rescaling and unnormalized advantage are activated. The initial weights are the MSE-pretrained checkpoint at the indicated step.

Target	Sequence	Length	Pocket length	Init. ckpt	Total steps (switch)
1SSH	GPPPAMPARPT	11	35	32,000	18,000 (6,000)
2R02	ELPYLTSLRSL	11	67	24,000	12,000 (6,000)
5UZW	NASAPV	6	87	36,000	14,000 (8,000)
6V63	TLKYPIEQGIVTNWDDMEK	19	131	36,000	4,000 (–)

128-target test split (Appendix Table 6), our reimplementation reaches a lower AAR (0.286) than the released PepFlow checkpoint (0.490), with correspondingly higher sequence diversity; retraining the original implementation under our conditions lands in the same range (pepflowww_repro, AAR 0.265). The architectural changes of Section 3 target the per-step likelihood that reinforcement learning requires rather than supervised reconstruction, so this is a parity check, not a claim of improvement. Agreement with the single crystallized sequence is also not the quantity we optimize: what GRPO needs from the initialization is reward variation within a sampled group. All comparisons below share this pretrained model. For each target, we select the checkpoint with the highest clash-free rate as the RL initialization (Table 1).

Targets. We fine-tune four complexes independently (Table 1), all drawn from the same 128-complex test split used above (Appendix Table 6). That split retains only complexes whose receptor is a single chain, so the pocket given to the model is always defined by one receptor chain and no multi-chain assembly has to be handled. 1SSH has low reward sparsity and serves as a lightweight validation target. The remaining three span peptide lengths 6–19, share low sequence homology, and differ in binding mode; all exhibit large RMSD at pretraining. When the reward distribution concentrates at its lower bound, we switch to exponential rescaling with unnormalized advantage (Appendix C). 6V63 uses the unnormalized advantage from the start, together with group-level weighting w_g, but its reward is never rescaled.

Baselines and evaluation. The baseline for each target is its MSE-pretrained initialization (Table 1), sampled with the same noise schedule used during RL for fair comparison. All models generate 64 samples at $N = 100$ steps from a fixed seed. The binding pocket follows PepFlow's definition (receptor residues within 10 Å of the reference peptide). Comparisons with the reproduced PepFlow baseline are in the Appendix. Metrics include *reference agreement* (AAR, RMSD, SSR (Kabsch & Sander, 1983; McGibbon et al., 2015), BSR, structural diversity based on TM-score (Zhang & Skolnick, 2005)), *physical validity* (clash-free rate, clash counts, bond lengths), and *reward* (median and top-10% mean of equation 10). As independent validation we report unrelaxed Rosetta ΔG_{bind} (Alford et al., 2017; Chaudhury et al., 2010) in Appendix F.4.

5.2 Main Results

Table 2 summarizes results. Reward medians improve on all targets (e.g. $-23.93 \rightarrow -0.41$ on 1SSH) and clash-free rates rise substantially ($39.1 \rightarrow 79.7\%$, $17.2 \rightarrow 60.9\%$, etc.). This is not a degenerate shortcut: bond lengths stay near the natural 1.33 Å, SSR is preserved, and structural diversity is nearly unchanged (Figure 1c; Appendix Figures 3–6). KL regularization keeps backbone geometry within the natural range. Rosetta ΔG_{bind} also improves on all targets (Appendix F.4).

6V63 is the hardest case: with 19 residues and a 131-token pocket, only 3.1% of pretrained samples are clash-free, and 4,000 RL steps raise this to 10.9%. Physical validity thus improves here as well, but with nearly all rewards at the lower bound the advantage carries little signal and progress is slow.

These aggregate statistics are complemented by residue-level inspection of individual complexes for all four targets (Appendix E.6). Comparing poses generated before and after RL, the number of protein–ligand contacts flagged as high-energy clashes either decreases or is unchanged on every target, and the number of rotatable bonds in poor torsional geometry stays within a few bonds of its pretrained value, fluctuating in both directions rather than increasing systematically. Since the reward contains no term penalizing intramolecular strain, this absence of a systematic increase

Table 2: Main results (mean over 64 samples per target). "Pretrained" denotes the fine-tuning initialization, sampled with the same noise-injection schedule used during RL training. R_{med} / R_{top10} are the median and top-10% mean of the reward equation 10. Rewards improve on all targets and the clash-free rate increases substantially. Rosetta validation results are in Appendix F.4.

Target	Method	Clash-free↑	R_{med} ↑	R_{top10} ↑	BSR↑	RMSD↓
1SSH	Pretrained	39.1%	−23.93	−4.90	0.976	2.63
	PepFM-TanRL	**79.7%**	**−0.41**	**+0.69**	0.717	2.97
2R02	Pretrained	48.4%	−23.93	−1.23	0.718	10.25
	PepFM-TanRL	**70.3%**	**−2.54**	**+0.36**	0.628	10.90
5UZW	Pretrained	17.2%	−26.32	−10.51	0.843	8.33
	PepFM-TanRL	**60.9%**	**−6.13**	**−0.12**	0.820	8.59
6V63	Pretrained	3.1%	−29.18	−20.99	0.758	20.77
	PepFM-TanRL	**10.9%**	**−27.77**	**−16.59**	0.766	20.58

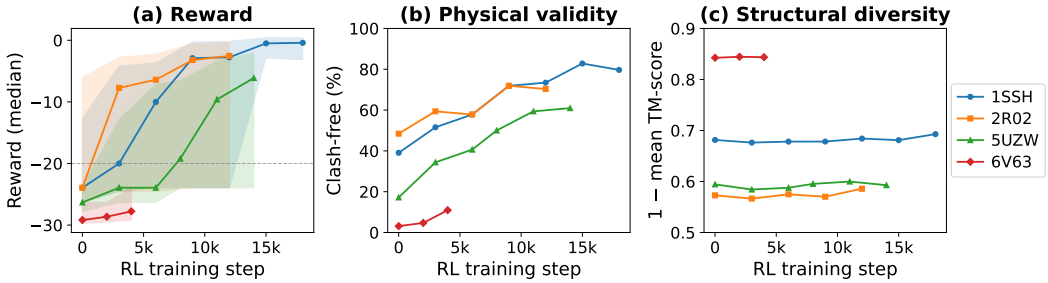


Figure 1: (a) Median reward over 64 evaluation samples (shaded band: 25th–75th percentiles). The dashed line marks the clash-penalty threshold $r_0 = -20$. Step 0 corresponds to the fine-tuning initialization. (b) Clash-free rate over 64 evaluation samples. (c) Structural diversity ($1 - \overline{\text{TM-score}}$; mean pairwise TM-score of the 64 C_α traces subtracted from 1). Diversity is largely preserved throughout RL training.

indicates that the improvement in intermolecular packing is not purchased at the cost of internal geometry: in no case did RL resolve clashes by distorting the peptide away from the pocket. As these observations rest on two selected structures per model and target, they should be read as a qualitative check rather than an ensemble-level statistic.

5.3 Similarity to the Reference Structure and Diversity

The reference structure is a single experimentally resolved example and does not necessarily represent the optimal binding pose, so changes in similarity under RL are expected. As shown in Table 2 (full checkpoint results in Appendix Table 5), RMSD increases slightly on most targets while AAR improves on some (e.g. 5UZW: $0.182 \rightarrow 0.302$). BSR drops on 1SSH ($0.976 \rightarrow 0.717$) and 2R02 but is preserved on the other targets. SSR is nearly unchanged on all targets, confirming that secondary structure is maintained. Structural diversity (Figure 1c) remains nearly constant (e.g. $0.681 \rightarrow 0.693$), indicating no mode collapse. Figure 2 shows that the penalty-region mass ($r < -20$) shifts into the clash-free range, and the reward distribution among clash-free samples also shifts toward higher Vinardo scores on all targets except 6V63. On some targets (e.g. 1SSH, 5UZW) the upper tail also improves over pretraining, but the rewards plateau at modest levels rather than reaching values expected of truly high-affinity binders. Overall, the policy suppresses energetically unfavorable conformations and produces more physically natural peptide structures while preserving diversity.

Visual inspection of the ensembles generated after RL supports the same conclusion at the level of individual structures (Appendix E.5). Clustering the 64 samples per target with $1 - \text{TM-score}$ as the

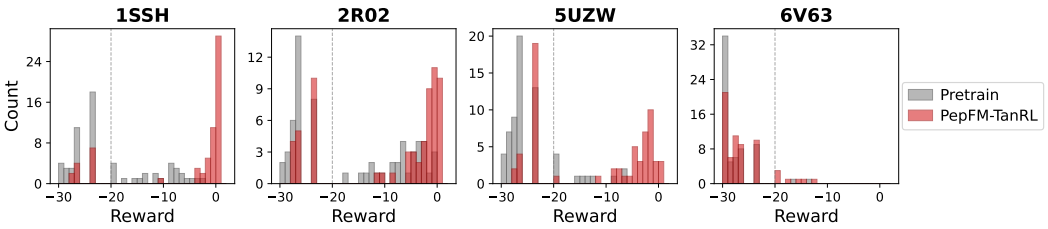


Figure 2: Reward distributions equation 10 for each target (64 samples). Gray: fine-tuning initialization; red: PepFM-TanRL. Rewards below -20 correspond to the clash-penalty region; RL training reduces this mass substantially. Among clash-free samples the distribution also shifts toward higher Vinardo scores, indicating that the policy generates more physically natural structures. On some targets the upper tail improves, but rewards plateau below levels expected of high-affinity binders.

distance metric and rendering the five cluster medoids shows that, on every target, the medoids retain the secondary structure and the binding orientation of the reference ligand while spanning diverse sequences, backbone traces, and side-chain presentation directions. The residual steric violations are target-dependent and interpretable: they are least frequent on the short helical targets and most frequent on the 19-residue target 6V63, where the peptide termini are least constrained by the pocket. Sequence diversity is likewise lowest on the 6-residue target 5UZW, where the small number of designable positions leaves little sequence entropy, yet its backbone conformations remain clearly varied across clusters.

Ablation experiments on the sequence parameterization and on the advantage normalization / reward rescaling strategy are presented in Appendix C. The `CE` initialization yields higher reward and clash-free rate than `MSE` at the same RL step count, and exponential reward rescaling with unnormalized advantage resolves the reward plateau observed in the later phase of training.

6 Discussion and Limitations

This work introduces a framework for applying GRPO to multi-modal flow models and demonstrates reward improvement across all four targets, confirmed by an independent scoring function (Rosetta ΔG_{bind}). Structural diversity is preserved and KL regularization prevents mode collapse (Section 5.3). However, Rosetta $\Delta G_{\text{bind}}^{\text{relaxed}}$ shows limited improvement (Appendix F.4), suggesting that improving post-relaxation performance requires evaluating rewards on relaxed structures.

Four limitations deserve mention. (1) Per-target fine-tuning requires thousands of steps; multi-target generalization is untested. (2) Where nearly all rewards sit at the floor (6V63), the advantage carries little signal and progress is slow. (3) Rosetta is a computational proxy, not an experimental measurement. (4) With four targets, statistical precision is limited.

7 Conclusion

We presented PepFM-TanRL, which injects time-decaying noise into a deterministic generative ODE to reinterpret a multi-modal flow model as a stochastic policy, deriving per-modality factored importance ratios and an analytical KL from closed-form tangent-space likelihoods. Joint torsion-angle prediction over all 20 amino acid types eliminates discrete sampling, so that each step's state transition has a closed-form Gaussian log-likelihood enabling importance-ratio-based RL. Reward distributions improve on all four targets, the clash-free rate rises from 39% to 80% on the best target, and unrelaxed Rosetta binding energy also improves. Structural diversity is preserved with no mode collapse. Future work includes addressing reward-floor targets, reward design for post-relaxation affinity, and multi-target generalization.

AI USE STATEMENT

Following the ICLR policy on AI use, we disclose the following. Among the uses that require disclosure, a generative AI assistant helped with translation while preparing the manuscript and with implementing methods (debugging and small utility routines written to author specification). The survey of related work, the formulation of hypotheses, the experimental design, and the interpretation of results were carried out by the authors; the assistant was used as a secondary check on these—surfacing prior work that might have been overlooked and flagging gaps in the reasoning—with every suggestion evaluated and accepted or rejected by the authors. It was not used to generate synthetic datasets, to clean or reformat data, to develop theoretical models or conceptual frameworks, or to formulate, prove, or supply ingredients for mathematical claims. Among the uses for which disclosure is recommended, it drafted and edited the text of the paper from the content and claims specified by the authors, identified and summarized related literature, formatted references, wrote plotting code for the figures, and assisted with brainstorming and information search; it was not used to suggest experimental parameters, to propose the structure of the paper, or to create other artifacts, and surveys, interviews, and transcription do not arise in this work. All AI-assisted output was reviewed by the authors: generated code was read and tested against the reported results, cited works were checked against their primary sources, and quantitative claims were verified against the implementation and the experiment logs. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a computational framework for peptide design and does not involve human subjects, animal experiments, or the collection of personal data. All experiments are conducted *in silico* using the publicly available preprocessed structural data released by PepFlow (Li et al., 2024). The generated peptide structures are computational proposals that require extensive experimental validation before any therapeutic application. We do not claim that the generated designs are ready for clinical use. The authors declare no conflicts of interest, and this work received no external funding.

REPRODUCIBILITY STATEMENT

The code and configuration files for pretraining, reinforcement learning, and evaluation, together with the scripts that aggregate the raw results into the tables of this paper, are provided to reviewers as anonymized supplementary material and will be released publicly upon acceptance. The dataset is the publicly available preprocessed data from PepFlow (Li et al., 2024). Training is distributed across a single node of 8×NVIDIA H200 GPUs (141 GB VRAM each) using Ray (Moritz et al., 2018). RL fine-tuning takes approximately two days per target. All hyperparameters in Table 4 match the accompanying configuration files.

REFERENCES

Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017. doi: 10.1021/acs.jctc.7b00125.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.13301.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.02391.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design.

In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5453–5512, 2024. arXiv:2402.04997.

Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J. Gray. PyRosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*, 26(5):689–691, 2010. doi: 10.1093/bioinformatics/btq007.

Chemical Computing Group ULC. *Molecular Operating Environment (MOE), 2024.06*. 910–1010 Sherbrooke St. West, Montreal, QC H3A 2R7, Canada, 2024.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2302.03660.

Tobiasz Ciepliński, Tomasz Danel, Sabina Podlewska, and Stanisław Jastrzębski. Generative models should at least be able to design molecules that dock well: A new benchmark. *Journal of Chemical Information and Modeling*, 63(11):3238–3247, 2023. doi: 10.1021/acs.jcim.2c01355.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.17400.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Rob J. de Haas, Neville Bethel, et al. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2409.08861.

Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, Kyle A. Beauchamp, Lee-Ping Wang, Andrew C. Simmonett, Matthew P. Harrigan, Chaya D. Stern, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):e1005659, 2017. doi: 10.1371/journal.pcbi.1005659.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. AutoDock vina 1.2.0: New docking methods, expanded force field, and python bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, 2021. doi: 10.1021/acs.jcim.1c00203.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 79858–79885, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866, 2023.

Wolfgang Guba, Agnes Meyder, Matthias Rarey, and Jérôme Hert. Torsion library reloaded: A new version of expert-derived SMARTS rules for assessing conformations of small molecules. *Journal of Chemical Information and Modeling*, 56(1):1–5, 2016. doi: 10.1021/acs.jcim.5b00522.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

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. doi: 10.1038/s41586-021-03819-2.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983. doi: 10.1002/bip.360221211.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-atom peptide design with geometric latent diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. arXiv:2402.13555.

Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-atom peptide design based on multi-modal flow matching. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27615–27640, 2024. arXiv:2406.00735.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel W. K. Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J. Hendel, Miriam K. Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using RoseTTAFold sequence space diffusion. *Nature Biotechnology*, 43(8):1288–1298, 2025. doi: 10.1038/s41587-024-02395-w.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025a. arXiv:2505.05470.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1711.05101.

Robert T. McGibbon, Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. MDTraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528–1532, 2015. doi: 10.1016/j.bpj.2015.08.015.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 561–577, 2018.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27730–27744, 2022.

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

Patrick Penner, Wolfgang Guba, Robert Schmidt, Agnes Meyder, Martin Stahl, and Matthias Rarey. The torsion library: Semiautomated improvement of torsion rules with SMARTScompare. *Journal of Chemical Information and Modeling*, 62(7):1644–1653, 2022. doi: 10.1021/acs.jcim.2c00043.

Rodrigo Quiroga and Marcos A. Villarreal. Vinardo: A scoring function based on AutoDock vina improves scoring, docking, and virtual screening. *PLOS ONE*, 11(5):e0155183, 2016. doi: 10.1371/journal.pone.0155183.

Philipp Renz, Dries Van Rompaey, Jörg Kurt Wegner, Sepp Hochreiter, and Günter Klambauer. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32–33:55–63, 2019. doi: 10.1016/j.ddtec.2020.09.003.

Christin Schärfer, Tanja Schulz-Gasch, Hans-Christian Ehrlich, Wolfgang Guba, Matthias Rarey, and Martin Stahl. Torsion angle preferences in druglike chemical space: A comprehensive guide. *Journal of Medicinal Chemistry*, 56(5):2016–2028, 2013. doi: 10.1021/jm3016816.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 9460–9471, 2022.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M. Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48892–48918, 2024. arXiv:2402.16359.

Gregory L. Warren, C. Webster Andrews, Anna-Maria Capelli, Brian Clarke, Judith LaLonde, Millard H. Lambert, Mika Lindvall, Neysa Nevins, Simon F. Semus, Stefan Senger, et al. A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry*, 49(20): 5912–5931, 2006. doi: 10.1021/jm050362n.

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. doi: 10.1038/s41586-023-06415-8.

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

Yang Zhang and Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005. doi: 10.1093/nar/gki524.

A Derivation of Importance Ratios and KL

A.1 Transition Kernel Log-Likelihood

Under the transition defined in equation 3, the next state for modality m and residue j is drawn from an isotropic Gaussian in the tangent space $T_{\mu_\theta^m}\mathcal{M}_m$ centered at the predicted mean μ_θ^m. Let d_m denote the tangent-space dimension ($\mathbb{R}^3$: 3; $\mathrm{SO}(3)$: 3; $\mathbb{T}$: 1 per angle; logit space: 20), and write the tangent vector to the realized next state $\mathcal{C}_{t+\Delta t}^{\mathrm{pep},m}$ as $u_{\theta,j}^m = \mathtt{log}_{\mu_\theta^m}(\mathcal{C}_{t+\Delta t}^{\mathrm{pep},m})_j$. Because the variance is $\sigma_m(t)^2\Delta t$,

$$\log \pi_\theta\big(\mathcal{C}_{t+\Delta t,j}^{\mathrm{pep},m} \mid \mathcal{C}_t^{\mathrm{pep}}\big) = -\frac{\|u_{\theta,j}^m\|^2}{2\sigma_m(t)^2\Delta t} - \frac{d_m}{2}\log\big(2\pi\sigma_m(t)^2\Delta t\big). \tag{11}$$

For $\mathrm{SO}(3)$ and $\mathbb{T}$ the transition kernel is defined by pushing forward a tangent-space Gaussian through the exponential map (i.e. $\mathcal{N}$ is a density on the tangent space), so the Jacobian arising from the manifold volume element is independent of θ. When $\sigma_m(t)\sqrt{\Delta t}$ is small relative to the diameter of $\mathrm{SO}(3)$ or the circumference of $\mathbb{T}$, wrapping contributions are negligible.

A.2 Importance Ratio

The reference policy π_{old} shares the same $\sigma_m(t)$ and Δt, differing only in the mean μ^m_{old}. The normalization constant (second term of equation 11) is therefore identical for both policies and cancels in the ratio. Defining the normalized noise $\xi^m_{\theta,j} = u^m_{\theta,j}/(\sigma_m(t)\sqrt{\Delta t})$,

$$\rho^m_j = \frac{\pi_\theta(\mathcal{C}^{\text{pep},m}_{t+\Delta t,j} \mid \mathcal{C}^{\text{pep}}_t)}{\pi_{\text{old}}(\mathcal{C}^{\text{pep},m}_{t+\Delta t,j} \mid \mathcal{C}^{\text{pep}}_t)} = \exp\left(\frac{\|u^m_{\text{old},j}\|^2 - \|u^m_{\theta,j}\|^2}{2\sigma_m(t)^2\Delta t}\right) = \exp\Big(\tfrac{1}{2}\big(\|\xi^m_{\text{old},j}\|^2 - \|\xi^m_{\theta,j}\|^2\big)\Big), \quad (12)$$

which yields equation 6. In practice ξ is computed directly. Although ξ can diverge near the terminal time where $\sigma_m(t) \to 0$, the same $\sigma_m(t)$ appears in numerator and denominator, keeping the ratio stable; residues outside the generation target are masked to $\xi = 0$.

For torsion angles, each residue j carries a 20×5 state. The conditional log-likelihood consistent with the probability-weighted feature construction is

$$\log \pi_\theta\big(\chi_{t+\Delta t,j} \mid \mathcal{C}^{\text{pep}}_t\big) = \sum_{a=1}^{20} p_t(a) \sum_{k=1}^{5} m_{a,k} \left[-\frac{\|u^{\text{ang}}_{\theta,j,a,k}\|^2}{2\sigma_{\text{ang}}(t)^2\Delta t} + \text{const}\right] \quad (13)$$

where $m_{a,k} \in \{0,1\}$ indicates whether amino acid a possesses a k-th torsion angle. The weighted form given in Section 4 follows directly.

A.3 KL to the Reference Policy

Both π_θ and π_{ref} are isotropic Gaussians with variance $\sigma_m(t)^2\Delta t$, differing only in their means. The KL divergence between d-dimensional isotropic Gaussians with equal variance is $\text{KL}(\mathcal{N}(\mu_1, \sigma^2 I) \,\|\, \mathcal{N}(\mu_2, \sigma^2 I)) = \|\mu_1 - \mu_2\|^2/(2\sigma^2)$; on the manifold the difference between means is replaced by the logarithmic map:

$$\text{KL}^m_j = \frac{\left\|\texttt{log}_{\mu^m_\theta}(\mu^m_{\text{ref}})_j\right\|^2}{2\sigma_m(t)^2\Delta t} \quad (14)$$

Averaging over valid residues gives equation 8. This form allows the KL to be evaluated analytically without sampling-based approximation. Note that as $\sigma_m(t) \to 0$ the denominator vanishes, so the effective weight of the KL grows near the terminal time. This penalizes deviations from the reference policy more strongly at later steps, which is a desirable property for keeping the generated final structures close to the pretrained distribution.

B Sequence Parameterizations

We consider two parameterizations for the sequence modality. In both cases the model architecture is identical: the network predicts the logit $\hat{s}_{1,\theta}$ at $t = 1$ from the state at time t, and the velocity is derived as $(\hat{s}_{1,\theta} - s_t)/(1-t)$.

The first (`MSE`) is the method described in the PepFlow paper. The target logit s_1 is a scaled one-hot vector in $\{-0.5, 9.5\}$ (sum zero, component gap 10). The loss minimizes the MSE between velocities derived from the prediction and the target:

$$\mathcal{L}^{\text{type}}_{\text{MSE}} = \mathbb{E}_{t,\mathcal{C}^{\text{pep}}_1}\left[\left\|\frac{\hat{s}_{1,\theta} - s_t}{1-t} - \frac{s_1 - s_t}{1-t}\right\|^2\right] = \mathbb{E}_{t,\mathcal{C}^{\text{pep}}_1}\left[\frac{\left\|\hat{s}_{1,\theta} - s_1\right\|^2}{(1-t)^2}\right] \quad (15)$$

Because a categorical distribution (logits at 0 and $-\infty$) is approximated with finite logits, the classification signal is weak.

The second (`CE`) is not described in the PepFlow paper but is adopted in its released implementation. It minimizes the cross-entropy between the predicted $t = 1$ logit $\hat{s}_{1,\theta}$ and the ground-truth category:

$$\mathcal{L}^{\text{type}}_{\text{CE}} = \mathbb{E}_{t,\mathcal{C}^{\text{pep}}_1}\Big[-\log \text{softmax}\big(\hat{s}_{1,\theta}(\mathcal{C}^{\text{pep}}_t, t)\big)[a_1]\Big] \quad (16)$$

Because the predicted logit does not target a specific value, the velocity field is not self-consistent. To reconcile this, the interpolation target is set to $\big(2\,\text{softmax}(\hat{s}_{1,\theta}) - 1\big)K$ ($K = 5$), rescaling to match the logit range of `MSE`, and integration proceeds via $s_{t+\Delta t} = s_t + \frac{\Delta t}{1-t}\big(\text{target} - s_t\big)$.

Table 3: Ablations on 1SSH (mean over 64 samples). Upper half: comparison of pretraining sequence parameterizations (6,000 RL steps, no reward rescaling). Lower half: comparison of continuation strategies beyond 6,000 steps.

Setting	Clash-free↑	R_{med} ↑	BSR↑	RMSD↓
Sequence parameterization at initialization (6,000 steps)				
`MSE` init. (pretrained)	39.1%	−23.93	0.976	2.63
`MSE` → PepFM-TanRL	57.8%	−10.00	0.942	2.52
`CE` init. (pretrained)	31.3%	−23.93	0.833	2.95
`CE` → PepFM-TanRL	**70.3%**	**−3.05**	0.836	**2.36**
Continuation from 6,000 to 12,000 steps (`MSE` init.)				
Original settings	57.8%	−10.00	0.942	2.52
Exp. rescaling + no std normalization	**73.4%**	**−2.77**	0.795	2.65

C Ablations

Sequence parameterization at pretraining. The upper half of Table 3 compares fine-tuning from the `MSE` initialization (32,000 pretraining steps) with the `CE` initialization (40,000 steps) on 1SSH under identical RL conditions (6,000 steps, before exponential rescaling). At the same RL step count, the `CE` initialization yields a higher reward median (-3.05 vs. -10.00) and clash-free rate (70.3% vs. 57.8%). Moreover, BSR slightly increases ($0.833 \rightarrow 0.836$) while RMSD simultaneously improves ($2.95 \rightarrow 2.36$ Å), suggesting that the choice of initialization influences the direction of reward optimization.

Advantage normalization and reward rescaling. The lower half of Table 3 compares two continuation strategies from the 6,000-step checkpoint on 1SSH: (i) continuing with the original settings, and (ii) switching to exponential rescaling with unnormalized advantage. Under the original settings, the reward plateaus at 6,000 steps and the median remains at -10.00 through 12,000 steps. Switching to the rescaled variant brings the median down to -2.77. When rewards concentrate at the lower bound, the within-group standard deviation shrinks and the division in equation 2 erases the distinction between "slightly better" and "much better." The exponential transform $r \mapsto \exp\big((r - r_{\text{ref}})/\tau_r\big)$ ($r_{\text{ref}} = -5.0$, $\tau_r = 4.0$) compresses differences in the low-reward region and amplifies them in the high-reward region; removing normalization lets these differences pass through, resolving the stall. This improvement, however, comes at the cost of lower BSR ($0.942 \rightarrow 0.795$), indicating that stronger reward optimization is accompanied by exploration away from the reference binding pose.

D Training Details

Each gradient-step batch consists of $32 \times 2 \times 4 = 256$ transitions per GPU (32 time points, 2 groups, 4 samples per group), accumulated over 4 batches before each update. Reward evaluation runs as Ray (Moritz et al., 2018) actors (eight workers per GPU), overlapping with sampling to hide the latency of docking computation. Policy updates use AdamW (Loshchilov & Hutter, 2019) with the settings in Table 4.

6V63 has a 19-residue peptide and a 131-token pocket, making both reward evaluation and policy updates substantially more expensive than for the other targets. Even with 141 GB VRAM (NVIDIA H200), a single GPU cannot accommodate two groups per batch; we therefore reduce the group count to 1 and increase gradient accumulation to 8. As a result, the number of RL steps achievable in the same training period ($\sim$2 days per target) is considerably lower (Table 1). Policy updates use DDP across 8×NVIDIA H200 GPUs (141 GB VRAM each), and RL fine-tuning takes approximately two days per target. Gradient norms are clipped at 100, and steps with non-finite gradients are skipped.

Four aspects of the actual runs deviate from Table 4. First, the peptide of 5UZW is only six residues long, so we narrow the residue-count perturbation to $[-1, +1]$ over its first 8,000 steps to keep the sampled peptides from becoming too short. Second, on 1SSH the reward plateaus again beyond 12,000 steps, and we loosen the KL constraint to $\beta = 10^{-4}$ for the remaining steps in order to

Table 4: Key hyperparameters of PepFM-TanRL. All values are taken directly from the configuration files used in the experiments and apply to every target unless a per-target deviation is noted below. Batch sizes are per-GPU; training uses 8 GPUs with DDP.

Parameter	Value	Parameter	Value
Sampling steps N	100	Optimizer	AdamW
Group size G	32	Learning rate	1×10^{-5}
Sampler batch size	128	Weight decay	1×10^{-3}
Train batch (time×group×sample)	32×2×4	Warmup steps	400
Gradient accumulation batches	4	Gradient norm clip	100
Buffer capacity (per GPU)	8 groups		
Epochs per buffer	64	$\sigma_{\text{pos}}^{\max}$	1.0
Clipping range ϵ	0.2	$\sigma_{\text{ori}}^{\max}, \sigma_{\text{ang}}^{\max}$	0.5
KL coefficient β	10^{-3}	$\sigma_{\text{type}}^{\max}$	5.0
$\lambda^{\text{pos}}, \lambda^{\text{ori}}$	0.5	Centroid shift	2.0 Å
$\lambda^{\text{ang}}, \lambda^{\text{type}}$	1.0	Residue count perturbation	$[-2, +2]$
$r_{\min}, r_0$	-20	Clash threshold d_{th}	1.1 Å
r_{worst}	-30	Search box	50^3 Å^3
κ_1, κ_2	0.5	Parameter count	10.45 M

let training continue to progress. Third, 6V63 uses the unnormalized advantage and the group-level weight w_g from the start, but its reward is left unscaled throughout training: it is the same reward equation 10 from which the other three targets also start. Fourth, the step budget in each configuration file is an upper bound at which a run halts automatically; in practice every run was stopped by the wall-clock budget (∼2 days per target), so the step counts in Table 1 are smaller than those limits.

E Additional Experiment Results

E.1 Results for All Checkpoints

Table 5 reports results for all RL checkpoints.

E.2 Pretrained Model Validation

Table 6 compares our pretrained models with the publicly released PepFlow checkpoint and our reproduction of the original implementation on a 128-target test split. The reproduction (`pepflowww_repro`) trains the original implementation on the same preprocessed data, step count, and batch size as ours, so that the two architectures can be compared under a matched training run; they reach comparable AAR (0.265 and 0.286). Since our architectural changes target the per-step likelihood needed for reinforcement learning (Section 3) rather than supervised reconstruction, parity here is the relevant outcome.

E.3 Comparison with the Reproduced Baseline

Table 7 compares `pepflowww_repro_oh`—our reproduction of the original PepFlow implementation trained for 40,000 steps on the same preprocessed data with batch size 32 per GPU—with our pretrained model. Both models are evaluated with the deterministic sampler (no noise injection) to match the original PepFlow evaluation protocol. The reproduction uses the one-hot sampling procedure described in the original PepFlow paper. On 1SSH, our model achieves substantially higher Clash-free (32.8% vs. 6.3%) and lower RMSD (2.58 vs. 7.31 Å), while on 2R02 Clash-free is also much higher (48.4% vs. 18.8%). On 6V63, both models have near-zero Clash-free (1.6% and 0.0%), confirming the inherent difficulty of this target regardless of implementation. On 5UZW, the reproduction has higher Clash-free (34.4% vs. 10.9%) and higher AAR (0.242 vs. 0.224), suggesting that model characteristics vary by target.

Table 5: Results for all RL checkpoints (mean over 64 samples each). `CE` denotes fine-tuning from a `CE`-pretrained model. "Pretrained" rows are the starting points for each target's fine-tuning.

Checkpoint	Clash-free	R_{med}	AAR	BSR	RMSD	R_{top10}
1SSH Pretrained (32,000)	39.1%	−23.93	0.419	0.976	2.63	−4.90
3,000	51.6%	−20.00	0.413	0.959	2.58	−0.97
6,000	57.8%	−10.00	0.409	0.942	2.52	−0.69
9,000	71.9%	−2.90	0.406	0.857	2.45	+0.95
12,000	73.4%	−2.77	0.378	0.795	2.65	+0.74
15,000	82.8%	−0.51	0.388	0.746	2.76	+0.86
18,000	79.7%	−0.41	0.405	0.717	2.97	+0.69
1SSH Pretrained CE (40,000)	31.3%	−23.93	0.472	0.833	2.95	−1.26
CE 3,000	68.8%	−2.54	0.466	0.843	2.58	+1.18
CE 6,000	70.3%	−3.05	0.480	0.836	2.36	+0.87
2R02 Pretrained (24,000)	48.4%	−23.93	0.223	0.718	10.25	−1.23
3,000	59.4%	−7.73	0.196	0.704	10.64	−0.28
6,000	57.8%	−6.40	0.266	0.677	10.71	−0.06
9,000	71.9%	−3.21	0.244	0.656	10.90	+0.37
12,000	70.3%	−2.54	0.284	0.628	10.90	+0.36
5UZW Pretrained (36,000)	17.2%	−26.32	0.182	0.843	8.33	−10.51
3,000	34.4%	−23.93	0.159	0.845	8.43	−3.96
6,000	40.6%	−23.93	0.315	0.841	8.37	−2.75
8,000	50.0%	−19.22	0.302	0.832	8.37	−2.01
11,000	59.4%	−9.61	0.315	0.818	8.42	−0.44
14,000	60.9%	−6.13	0.302	0.820	8.59	−0.12
6V63 Pretrained (36,000)	3.1%	−29.18	0.173	0.758	20.77	−20.99
2,000	4.7%	−28.65	0.172	0.756	20.41	−21.12
4,000	10.9%	−27.77	0.156	0.766	20.58	−16.59

Table 6: Pretrained model validation (128 targets from `test_names.txt` in the preprocessed dataset distributed by PepFlow, excluding targets with non-standard amino acids and targets whose receptor consists of more than one chain, so that the pocket is always defined by a single receptor chain; mean over 64 samples per target; deterministic sampler without noise injection). Clash-free is the fraction of samples free of steric clashes. `pepflowww_orig` is the publicly released PepFlow checkpoint; `pepflowww_repro` reproduces the original implementation on the same preprocessed data, trained for 40,000 steps with batch size 32 per GPU as reported in the PepFlow paper. The bottom two rows are models trained with our architecture. Rows without the `_oh` suffix use the softmax probabilities of the predicted $t = 1$ logit directly, rescaled to $\{K, -K\}$, as the interpolation target—the same velocity-field transition as Ours (`CE`) described in Section 3. The suffix `_oh` instead samples from the softmax distribution to obtain a one-hot vector before rescaling (the procedure described in the original PepFlow paper), which introduces discrete sampling that prevents likelihood tracking and is therefore not used in this work.

Model	AAR↑	RMSD↓	SSR↑	BSR↑	Diversity	Clash-free↑
`pepflowww_orig`	0.490	6.42	0.903	0.844	0.487	27.1%
`pepflowww_orig_oh`	0.488	6.42	0.903	0.844	0.487	27.9%
`pepflowww_repro`	0.265	6.43	0.778	0.819	0.704	15.0%
`pepflowww_repro_oh`	0.265	6.44	0.778	0.819	0.704	15.9%
Ours (`MSE`)	0.286	7.74	0.834	0.787	0.630	16.2%
Ours (`CE`)	0.280	6.89	0.815	0.793	0.669	19.2%

Table 7 also includes results with the noise-injected stochastic sampler used during RL training ("+noise"). Noise injection tends to slightly improve the clash-free rate, while AAR, BSR, and RMSD remain comparable to the deterministic sampler, confirming that the injected noise does not substantially degrade generation quality.

Table 7: Comparison between the reproduced PepFlow baseline (`pepflowww_repro_oh`; trained on the same preprocessed data with the original paper's one-hot sampling procedure) and our pretrained model (`MSE`, the initialization for fine-tuning on each target); means over 64 samples per target. The top two rows per target use the deterministic sampler (no noise); "+noise" uses the stochastic sampler with the same noise schedule as RL training.

Target	Model	Clash-free	R_{med}	AAR	BSR	RMSD
1SSH	`pepflowww_repro_oh`	6.3%	−28.65	0.376	0.957	7.31
	Ours (`MSE`)	32.8%	−23.93	0.425	0.978	2.58
	Ours (`MSE`) +noise	39.1%	−23.93	0.419	0.976	2.63
2R02	`pepflowww_repro_oh`	18.8%	−27.77	0.178	0.888	6.84
	Ours (`MSE`)	48.4%	−23.93	0.217	0.723	10.32
	Ours (`MSE`) +noise	48.4%	−23.93	0.223	0.718	10.25
5UZW	`pepflowww_repro_oh`	34.4%	−23.93	0.242	0.833	8.37
	Ours (`MSE`)	10.9%	−26.32	0.224	0.854	8.39
	Ours (`MSE`) +noise	17.2%	−26.32	0.182	0.843	8.33
6V63	`pepflowww_repro_oh`	0.0%	−29.76	0.201	0.800	5.89
	Ours (`MSE`)	1.6%	−28.91	0.167	0.761	20.82
	Ours (`MSE`) +noise	3.1%	−29.18	0.173	0.758	20.77

Table 8: Structural details (mean over 64 samples). Intra / Inter denote intra-peptide and peptide–receptor clash counts (number of heavy-atom pairs closer than 1.1 Å). Bond_{med} / $\text{Bond}_{\max}$ are the median and maximum peptide bond lengths (Å) between adjacent residues.

Target	Method	Intra↓	Inter↓	Bond_{med}	$\text{Bond}_{\max}$
1SSH	Pretrained	0.39	0.94	1.297	1.488
	PepFM-TanRL	**0.22**	**0.09**	1.336	1.599
2R02	Pretrained	0.33	0.88	1.357	1.520
	PepFM-TanRL	**0.23**	**0.27**	1.354	1.558
5UZW	Pretrained	1.05	0.98	1.293	1.537
	PepFM-TanRL	**0.34**	**0.17**	1.350	1.587
6V63	Pretrained	1.13	3.89	1.389	1.699
	PepFM-TanRL	**0.73**	**2.73**	1.390	1.694
1SSH (CE)	Pretrained	0.94	0.20	1.243	1.505
	PepFM-TanRL	**0.34**	**0.05**	1.401	1.625

E.4 Structural Details

Table 8 reports clash counts and peptide bond lengths. RL fine-tuning preferentially reduces peptide–receptor clashes: from $0.94 \rightarrow 0.09$ on 1SSH and from $0.98 \rightarrow 0.17$ on 5UZW. Intra-peptide clashes also decrease, albeit by a smaller margin. This asymmetry arises because peptide–receptor clashes can be resolved by adjusting the rigid-body placement alone and are thus easier to learn, even though the reward equation 10 weights both clash types equally ($\kappa_1 = \kappa_2 = 0.5$). Median peptide bond lengths shift slightly from 1.29–1.39 Å (pretrained) to 1.34–1.39 Å, remaining close to the canonical value of 1.33 Å and confirming that reward optimization does not disrupt backbone covalent geometry.

E.5 Structural Diversity Analysis

Visualization protocol. To assess whether the diversity of the generated ensembles survives RL fine-tuning, we visually inspected representative structures for all four targets (Figures 3–6). For each target, the 64 peptides generated after GRPO fine-tuning were hierarchically clustered using $1 - \text{TM-score}$ as the distance metric, the ensemble was partitioned into five clusters, and the medoid of each cluster was retained as its representative. All molecular graphics were prepared with

MOE 2024 (Chemical Computing Group ULC, 2024); the receptor is drawn as a molecular surface colored by surface property (magenta: polar; green: hydrophobic; red: exposed), the ground-truth peptide in dark gray, and the five medoids in the colors listed in panel (c) of each figure. Our generative model writes each sample as a PDB file containing atomic Cartesian coordinates only, without explicit connectivity records; the covalent topology is therefore perceived by the viewer from interatomic distances. As a consequence, MOE occasionally assigned an incorrect bond topology to the generated peptides: bonds were sometimes abnormally short or long, and intramolecular side-chain pairs that came into close contact were sometimes perceived as bonded. In such cases we manually corrected the affected bonds so that each peptide carried the correct covalent topology before rendering. This correction affects visualization only and was not applied to any structure used for the quantitative evaluations. Below, "bump" denotes a visually evident severe steric overlap, i.e., an atom or residue fragment buried inside the receptor surface. Since residue numbers are not displayed in Figures 3–6, residues are referred to by their position along the peptide chain, indexed from 1 at the N-terminus; this differs from the residue labels used in Appendix E.6, which follow the numbering of the generated PDB files.

1SSH (Figure 3). The native ligand of 1SSH is an 11-residue linear peptide adopting an extended, essentially straight conformation along the binding groove. All five medoids reproduce this extended secondary structure and the N-to-C orientation of the ground-truth peptide, while their sequences differ substantially from one another and from the native sequence (panel (c)); the backbone traces and the directions in which side chains are projected toward the receptor vary appreciably across clusters. The only severe steric violation is the side chain of Arg10 in Cluster 5, which is buried in the receptor surface; apart from this single residue, no severe backbone or side-chain bumps are observed in any of the five representatives.

2R02 (Figure 4). The native ligand of 2R02 is also an 11-residue linear peptide, but it binds as an α-helix. The generated medoids recapitulate the native secondary structure and binding orientation—all five representatives are helical and are threaded through the groove in the same direction as the ground truth—while displaying diverse sequences, helical backbone geometries, and side-chain presentation patterns. Severe bumps are limited to four side chains: Lys2 and Leu6 of Cluster 1, Asn9 of Cluster 4, and Lys3 of Cluster 5, each of which penetrates the receptor surface.

5UZW (Figure 5). The native ligand of 5UZW is a short, 6-residue linear peptide with an undulating (wave-like) backbone. The generated medoids follow the native secondary structure and orientation closely, and—in contrast to the other three targets—their sequences are comparatively conserved, with several clusters sharing residues at multiple positions. Even so, the backbone conformations and side-chain orientations remain clearly diverse across clusters, indicating that structural diversity is retained even when sequence diversity is reduced. Severe bumps are restricted to the side chains of His2 and Ser5 in Cluster 5, which are buried in the receptor surface; all other residues in all five representatives are free of severe overlaps.

6V63 (Figure 6). The native ligand of 6V63 is a long, 19-residue linear peptide that wraps around the receptor in a C-shaped curve following the protein surface. The generated medoids reproduce this overall curvature and binding orientation, and, as for 1SSH and 2R02, they exhibit substantial sequence diversity together with diverse backbone traces and side-chain presentation. However, this target is markedly harder to generate cleanly, and severe steric violations are considerably more frequent. Side-chain bumps are observed for Leu2, Ile4, Leu7, Asp8 and Phe9 (Cluster 1), Phe14 (Cluster 2), Met1, Leu2, Ser5, Pro13 and Ser14 (Cluster 3), Asp1, Leu2, Ser5 and Arg9 (Cluster 4), and Ile3 (Cluster 5). In addition, several residues show severe backbone overlaps with the receptor: Met1 and Ser5 of Cluster 3, Asp1 and Leu2 of Cluster 4, and Met1 of Cluster 5.

Summary. The visual inspection supports two conclusions. First, RL fine-tuning does not collapse the generative distribution onto a single mode: for every target, the five cluster medoids preserve the secondary structure and the binding orientation of the native ligand while spanning diverse sequences, backbone conformations, and side-chain presentation directions. This is the structural counterpart of the diversity metrics reported in Section 5.3, and it indicates that the reward-driven suppression of energetically unfavorable conformations acts on steric quality rather than on conformational or sequence variety. Second, the residual failure modes are target-dependent and interpretable. For the short 6-residue target (5UZW) the sequence diversity of the generated ensemble is noticeably lower than for the other targets, presumably because the small number of designable positions and the tight pocket leave little sequence entropy. For the long 19-residue target (6V63)

generation is clearly more difficult, and both the number and the severity of bumps increase. Across all targets, backbone bumps occur almost exclusively near the peptide termini—where the chain is least constrained by the pocket and where the reward provides the weakest per-residue signal—while side-chain bumps are strongly biased toward residues with long or bulky side chains (Arg, Lys, Leu, Ile, Met, Phe). These observations suggest that terminus-aware and side-chain-length-aware clash penalties, or a short post-generation side-chain repacking step, are natural directions for further reducing the remaining steric violations.

E.6 Structural Validity Analysis

Visualization protocol. While Appendix E.5 examined the diversity of the generated ensembles, this section asks a complementary question: does the reward-driven removal of steric clashes come at the price of internal geometric quality? A policy penalized for protein–ligand overlap could, in principle, reduce that overlap by adopting internally strained conformations that escape the pocket and forgo favorable contacts. To test this, we compare structures generated by the pretrained model with those generated after GRPO fine-tuning in terms of both torsional strain and protein–ligand steric clashes (Figures 7–10).

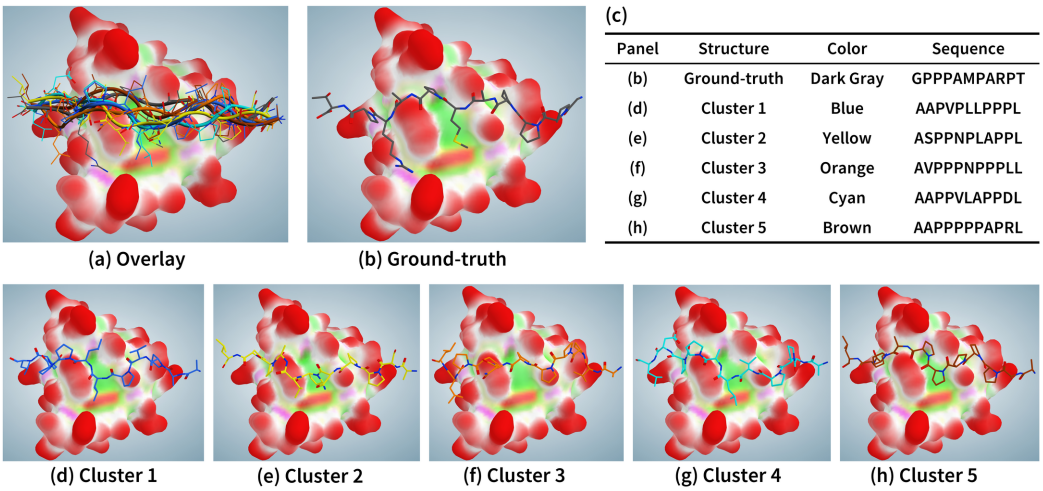
For each of the four targets, two structures were selected from the 64 samples generated by each model: (i) the structure with the best Vinardo score, taken as the most energetically favorable pose, and (ii) the structure most distant from (i) in TM-score, taken as a representative of a structurally dissimilar mode. Examining these two extremes lets us verify that any improvement is not confined to the single best-scoring sample.

Torsional strain of rotatable bonds was assessed with the Ligand Torsions tool in MOE 2024 (Chemical Computing Group ULC, 2024), which compares each rotatable-bond dihedral against the Torsion Library derived from small-molecule crystal-structure statistics (Schärfer et al., 2013; Guba et al., 2016; Penner et al., 2022) and color-codes bonds as good (green), warning (orange), or bad (red). Protein–ligand steric clashes are drawn as yellow dashed lines, with yellow disks whose size reflects the clash energy. Below, "bad torsion" refers to a rotatable bond flagged red, and "clash" to a contact rendered by MOE. Unlike Appendix E.5, residues are identified here by the numbering assigned in the generated PDB files, so that all residue labels correspond directly to those displayed in Figures 7–10.

Two caveats apply to the counts reported below. First, as in Appendix E.5, our model writes each sample as a PDB file without explicit connectivity records, so MOE must perceive the covalent topology from interatomic distances; where bonds were perceived incorrectly, we manually corrected them before rendering (visualization only; not applied to quantitative evaluations). Second, the clash criterion used here differs from the one used in the reward: the reward counts a clash whenever a protein–ligand heavy-atom distance falls below 1.1 Å, whereas MOE displays only contacts whose clash energy exceeds 10 kcal/mol. The clash counts in Figures 7–10 therefore do not coincide with the clash statistics optimized during training, and should be read as an independent, energy-based view of the same phenomenon.

1SSH (Figure 7). For the pretrained model, the best-Vinardo-score structure (panel (c)) contains five bad torsions and clashes at Arg5, Pro8, Pro10 and Arg12, while the most distant structure (panel (d)) contains a single bad torsion with clashes at Pro4, Leu7 and Phe11. After RL, the best-Vinardo-score structure (panel (e)) contains two bad torsions and no displayed clash at all, and the most distant structure (panel (f)) retains a single bad torsion with a clash only at Pro9. RL therefore reduces both the number of severely strained torsions in the best-scoring pose (five to two) and the number of clashing residues (four to zero; three to one), while the peptide remains packed along the binding groove.

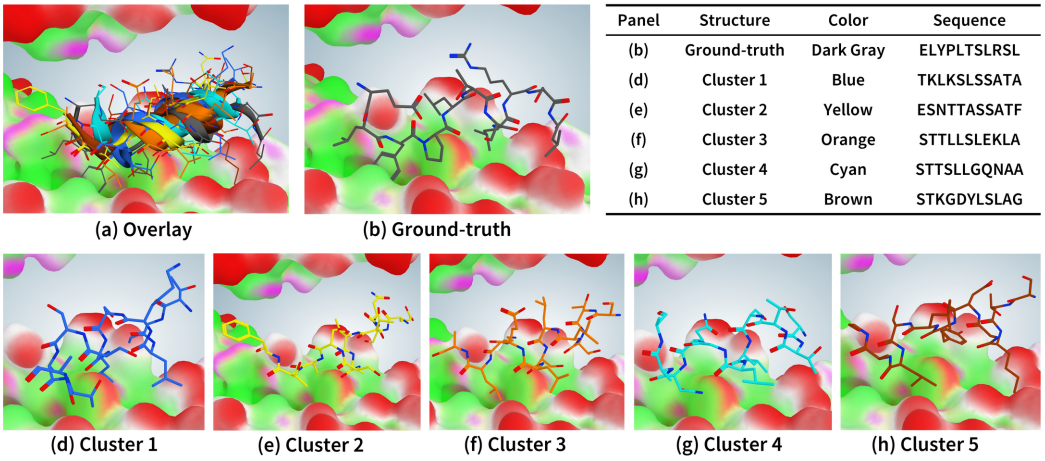
2R02 (Figure 8). For this helical target, the pretrained structures are already largely valid: panel (c) contains two bad torsions with a clash at Ala42, and panel (d) a single bad torsion with a clash at Leu42. After RL, panel (e) contains three bad torsions with a clash at Leu40, and panel (f) five bad torsions with a clash at Leu39. Each of the four structures thus carries exactly one displayed clash, and the number of bad torsions remains in the same low single-digit range. Because the bound conformation is an α-helix, the accessible conformational space is intrinsically narrower than for the extended peptides, and the pretrained model already produces predominantly valid poses.



(c)

Panel	Structure	Color	Sequence
(b)	Ground-truth	Dark Gray	GPPPAMPARPT
(d)	Cluster 1	Blue	AAPVPLLPPPL
(e)	Cluster 2	Yellow	ASPPNPLAPPL
(f)	Cluster 3	Orange	AVPPPNPPPLL
(g)	Cluster 4	Cyan	AAPPVLAPPDL
(h)	Cluster 5	Brown	AAPPPPPAPRL

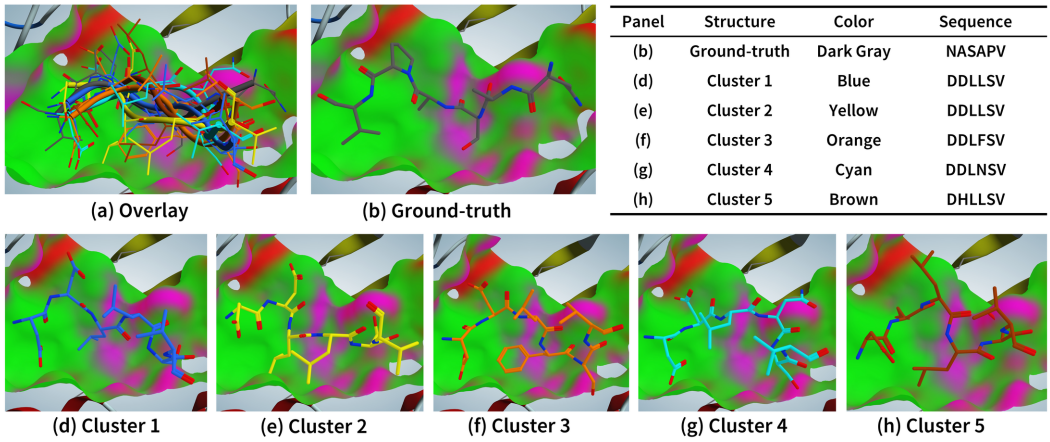
Figure 3: Structural diversity of generated peptides after RL for 1SSH. The 64 peptides were hierarchically clustered into five groups (distance $= 1 - \text{TM-score}$), and each cluster medoid is shown within the binding pocket. (a) Overlay of the ground-truth peptide and the five cluster medoids. The binding pocket is shown as a molecular surface visualized in MOE, colored by surface property (magenta: polar; green: hydrophobic; red: exposed). (b) Ground-truth peptide (dark gray). (c) Panel-to-cluster correspondence, listing the color and amino-acid sequence for each structure. (d–h) The five cluster medoids (Clusters 1–5).



(c)

Panel	Structure	Color	Sequence
(b)	Ground-truth	Dark Gray	ELYPLTSLRSL
(d)	Cluster 1	Blue	TKLKSLSSATA
(e)	Cluster 2	Yellow	ESNTTASSATF
(f)	Cluster 3	Orange	STTLLSLEKLA
(g)	Cluster 4	Cyan	STTSLLGQNAA
(h)	Cluster 5	Brown	STKGDYLSLAG

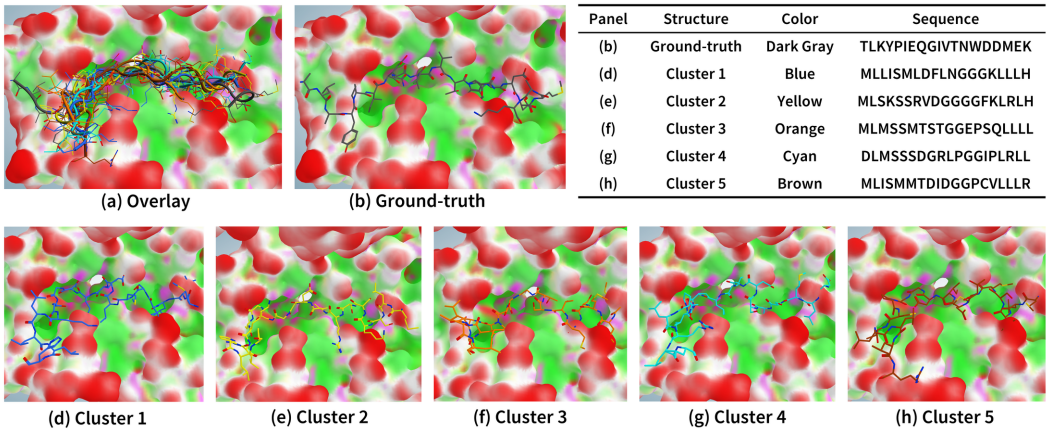
Figure 4: Structural diversity of generated peptides after RL for 2R02. The 64 peptides were hierarchically clustered into five groups (distance $= 1 - \text{TM-score}$), and each cluster medoid is shown within the binding pocket. (a) Overlay of the ground-truth peptide and the five cluster medoids. The binding pocket is shown as a molecular surface visualized in MOE, colored by surface property (magenta: polar; green: hydrophobic; red: exposed). (b) Ground-truth peptide (dark gray). (c) Panel-to-cluster correspondence, listing the color and amino-acid sequence for each structure. (d–h) The five cluster medoids (Clusters 1–5).



(c)

Panel	Structure	Color	Sequence
(b)	Ground-truth	Dark Gray	NASAPV
(d)	Cluster 1	Blue	DDLLSV
(e)	Cluster 2	Yellow	DDLLSV
(f)	Cluster 3	Orange	DDLFSV
(g)	Cluster 4	Cyan	DDLNSV
(h)	Cluster 5	Brown	DHLLSV

Figure 5: Structural diversity of generated peptides after RL for 5UZW. The 64 peptides were hierarchically clustered into five groups (distance $= 1 - \mathrm{TM\text{-}score}$), and each cluster medoid is shown within the binding pocket. (a) Overlay of the ground-truth peptide and the five cluster medoids. The binding pocket is shown as a molecular surface visualized in MOE, colored by surface property (magenta: polar; green: hydrophobic; red: exposed). (b) Ground-truth peptide (dark gray). (c) Panel-to-cluster correspondence, listing the color and amino-acid sequence for each structure. (d–h) The five cluster medoids (Clusters 1–5).



(c)

Panel	Structure	Color	Sequence
(b)	Ground-truth	Dark Gray	TLKYPIEQGIVTNWDDMEK
(d)	Cluster 1	Blue	MLLISMLDFLNGGGKLLLH
(e)	Cluster 2	Yellow	MLSKSSRVDGGGGFKLRLH
(f)	Cluster 3	Orange	MLMSSMTSTGGEPSQLLLL
(g)	Cluster 4	Cyan	DLMSSSDGRLPGGIPLRLL
(h)	Cluster 5	Brown	MLISMMTDIDGGPCVLLLR

Figure 6: Structural diversity of generated peptides after RL for 6V63. The 64 peptides were hierarchically clustered into five groups (distance $= 1 - \mathrm{TM\text{-}score}$), and each cluster medoid is shown within the binding pocket. (a) Overlay of the ground-truth peptide and the five cluster medoids. The binding pocket is shown as a molecular surface visualized in MOE, colored by surface property (magenta: polar; green: hydrophobic; red: exposed). (b) Ground-truth peptide (dark gray). (c) Panel-to-cluster correspondence, listing the color and amino-acid sequence for each structure. (d–h) The five cluster medoids (Clusters 1–5).

There is accordingly little headroom for RL to improve structural validity, and the fine-tuned policy essentially preserves the geometric quality already attained during pretraining.

5UZW (Figure 9). This target is intrinsically clash-prone: even the ground-truth complex (panel (a)) exhibits a displayed clash at Val32, showing that the 10 kcal/mol criterion flags contacts present in the crystal structure itself. Among the generated structures, panel (c) contains two bad torsions with clashes at Asp28, Leu29, Asn31 and Lys32, and panel (d) four bad torsions with clashes at His27, Asp28, Leu29 and Asp31. After RL, panel (e) contains four bad torsions with clashes reduced to Asp27 and Ser31, whereas panel (f) contains four bad torsions with clashes at Asp28, Leu29, Ser31 and Val32. RL thus halves the number of clashing residues in the best-scoring pose (four to two) but leaves the dissimilar mode unchanged in count. Given that the native ligand itself clashes under this criterion, and that the binding site is small and shallow, we regard a residual clash count of this magnitude as close to the practical floor for this target.

6V63 (Figure 10). This 19-residue target is the hardest of the four, and even the ground-truth ligand (panel (a)) contains two bad torsions. For the pretrained model, panel (c) contains eight bad torsions with clashes at Met66, Leu67, Lys70, Ser73, His75, Val76, Gly77 and Leu82, and panel (d) eleven bad torsions with clashes at Met66, Leu67, Ser69, Lys71, Lys72, Asp74, Ile75, Arg79 and Leu83. After RL, panel (e) contains six bad torsions with clashes at Met66, Pro68, Ser70, Gly75, Gly76, His78 and Met81, and panel (f) thirteen bad torsions with clashes at Leu67, Asp70, Val75, Leu76, Gly78, Leu79 and Leu81. The number of clashing residues decreases in both selected poses (eight to seven, and nine to seven), and the number of bad torsions decreases in the best-scoring pose (eight to six), although the most distant RL sample remains highly strained (eleven to thirteen). Consistent with the analysis of Figure 6, the severe clashes are concentrated toward the peptide termini, where the chain is least constrained by the receptor surface.

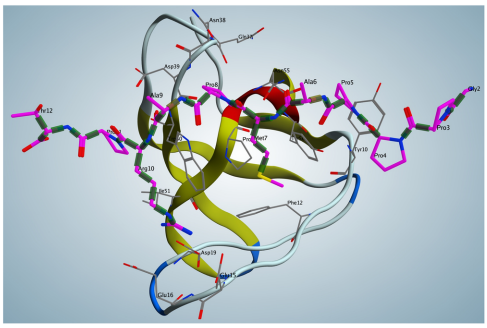
Summary. Across all four targets, RL fine-tuning does not induce the degenerate solution one might fear from a clash-penalized reward. In no case did the fine-tuned policy generate a peptide that reduces steric overlap by distorting away from the pocket and abandoning its interactions; every RL sample remains packed along the binding groove. Instead, the number of high-energy clashes decreases—completely for 1SSH (four clashing residues to none in the best-scoring pose), by half for 5UZW, modestly but consistently for 6V63, and negligibly for the already-clean helical target 2R02—while the number of severely strained torsions is broadly preserved, remaining within a few bonds of the pretrained level and in fact decreasing in the best-scoring poses of 1SSH and 6V63. The reward improves intermolecular packing without trading it against intramolecular geometry. Two target-specific limits are worth noting: for 5UZW the native complex itself is flagged as clashing, so the achievable clash count is bounded from below by the tightness of the pocket, and for 6V63 the length of the peptide leaves terminal residues under-constrained, which remains the dominant residual failure mode. These observations are based on two selected structures per model and target and are intended as a qualitative, per-atom complement to the ensemble-level clash and reward statistics reported in Section 5.2.

F Model Architecture Details

Algorithms 3–5 detail the model architecture. Node and edge embeddings (h_0, z_0) are pre-computed from the receptor $\mathcal{C}^{\text{rec}}$ once before generation (Algorithm 1); peptide token slots are allocated but initialized to blank, ensuring no leakage of information about the peptide state. The model forward pass (Algorithm 3) fuses side-chain features with h_0 and passes them through L FoldIteration blocks consisting of IPA (Jumper et al., 2021), Transformer self-attention, and node transition layers. Output heads predict amino acid logits (20 classes) at $t = 1$ and 20×5 torsion angles; backbone transforms are updated via BackboneUpdate (Jumper et al., 2021) in each block, applied only to peptide residues while keeping the receptor unchanged.

F.1 Time Conditioning via AdaLN-Zero

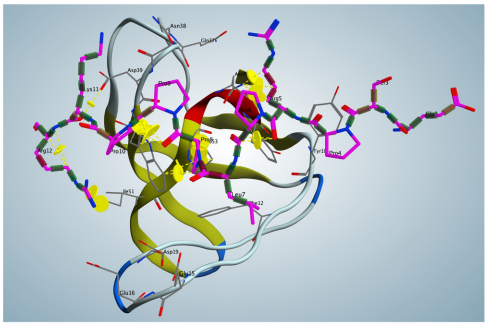
To effectively inject time information into the network, we adopt Adaptive Layer Normalization with zero-initialized gates (AdaLN-Zero) (Peebles & Xie, 2023) across all modules. In each of the IPA, Transformer self-attention, and node transition layers, a linear projection of the time embedding e_t produces scale γ, shift β, and gate g parameters. The gate weights are initialized to zero, so that



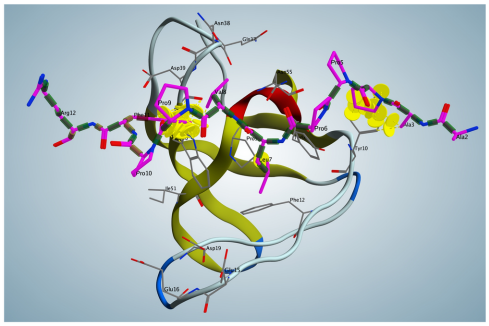
(a) Ground-truth

(b)

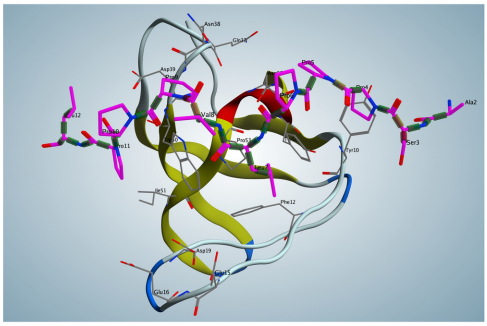
Panel	Sequence
(a)	GPPPAMPARPT
(c)	ESPRPLPPPKR
(d)	AAPPPLVPPFR
(e)	ASPPPLVPPPL
(f)	AVPPPPNPPLL



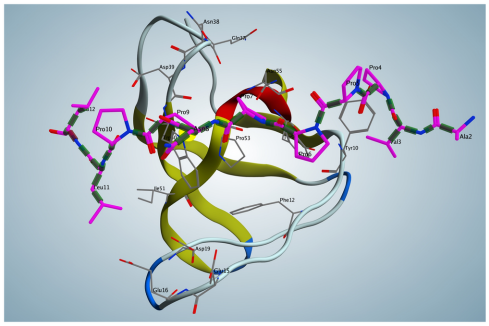
(c) Pretrained, best Vina score



(d) Pretrained, most distant

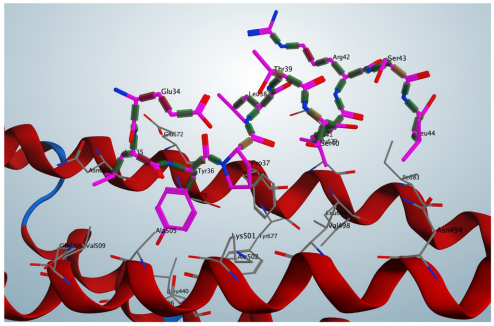


(e) RL, best Vina score



(f) RL, most distant

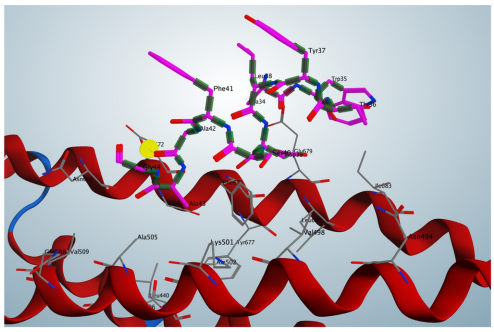
Figure 7: Structural validity of generated peptides for 1SSH. Ligands are shown in magenta. Torsional strain of rotatable bonds is assessed using the Ligand Torsions tool in MOE, where cylinders along rotatable bonds are color-coded by geometric quality (green: good; orange: warning; red: bad). Protein–ligand steric clashes are indicated by yellow dashed lines; yellow disks denote the clash energy magnitude (threshold: 10 kcal/mol). (a) Ground-truth peptide. (b) Panel-to-sequence correspondence. (c, d) Structures from the pretrained model. (e, f) Structures after RL. (c, e) Best Vinardo score. (d, f) Most distant from the best-Vinardo-score structure in TM-score.



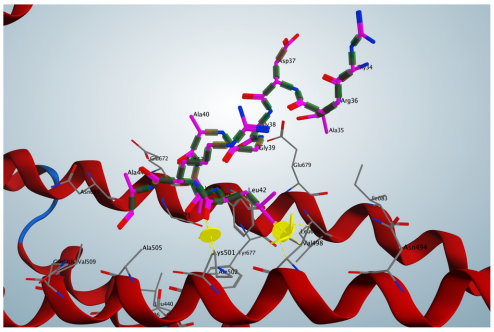
(a) Ground-truth

(b)

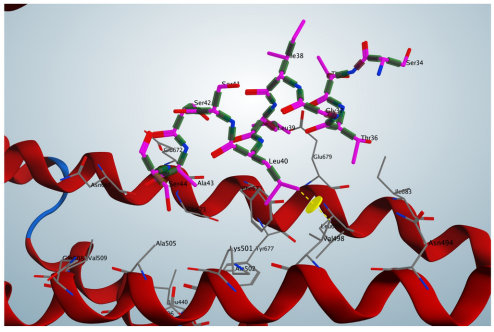
Panel	Sequence
(a)	ELYPLTSLRSL
(c)	AWTYLDSFAAG
(d)	GARDGGAVLRA
(e)	STTGILLSSAS
(f)	AATLTLSQGAA



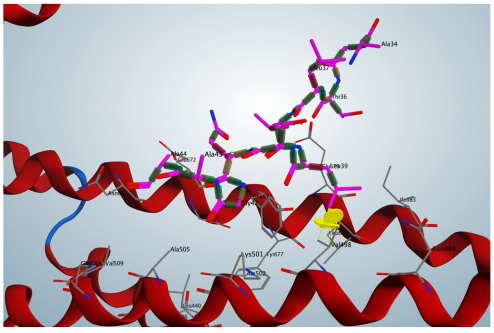
(c) Pretrained, best Vina score



(d) Pretrained, most distant

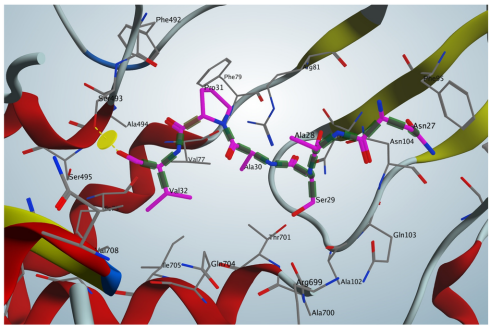


(e) RL, best Vina score



(f) RL, most distant

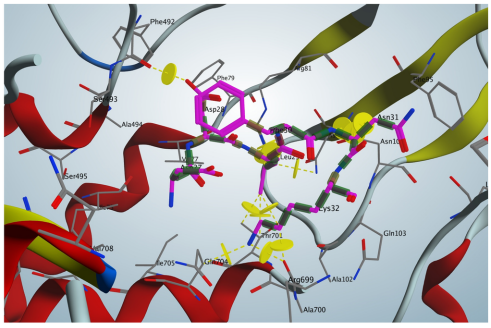
Figure 8: Structural validity of generated peptides for 2R02. Ligands are shown in magenta. Torsional strain of rotatable bonds is assessed using the Ligand Torsions tool in MOE, where cylinders along rotatable bonds are color-coded by geometric quality (green: good; orange: warning; red: bad). Protein–ligand steric clashes are indicated by yellow dashed lines; yellow disks denote the clash energy magnitude (threshold: 10 kcal/mol). (a) Ground-truth peptide. (b) Panel-to-sequence correspondence. (c, d) Structures from the pretrained model. (e, f) Structures after RL. (c, e) Best Vinardo score. (d, f) Most distant from the best-Vinardo-score structure in TM-score.



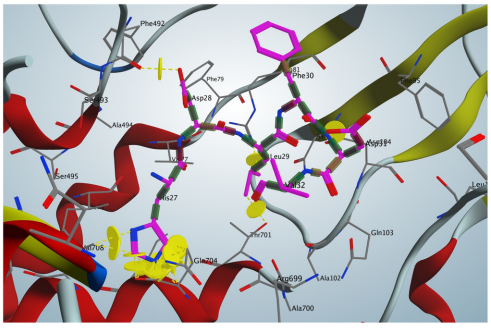
(a) Ground-truth

(b)

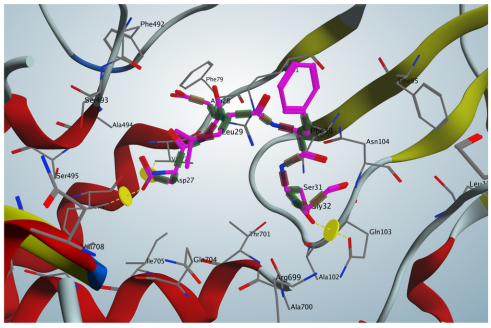
Panel	Sequence
(a)	NASAPV
(c)	DDLFNK
(d)	HDLFDV
(e)	DDLFSG
(f)	DDLLSV



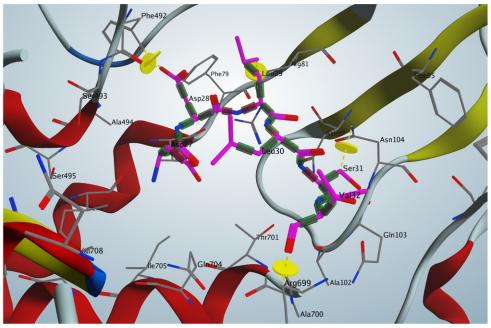
(c) Pretrained, best Vina score



(d) Pretrained, most distant

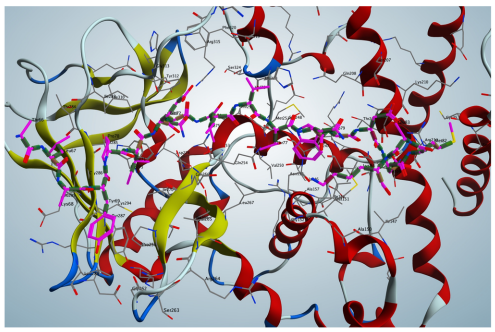


(e) RL, best Vina score



(f) RL, most distant

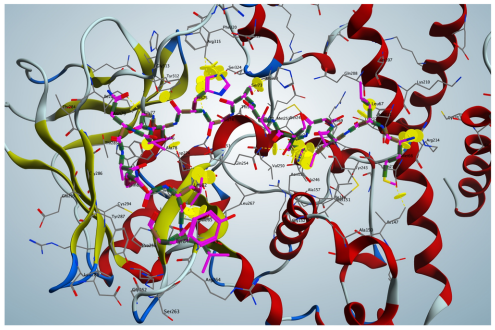
Figure 9: Structural validity of generated peptides for 5UZW. Ligands are shown in magenta. Torsional strain of rotatable bonds is assessed using the Ligand Torsions tool in MOE, where cylinders along rotatable bonds are color-coded by geometric quality (green: good; orange: warning; red: bad). Protein–ligand steric clashes are indicated by yellow dashed lines; yellow disks denote the clash energy magnitude (threshold: 10 kcal/mol). (a) Ground-truth peptide. (b) Panel-to-sequence correspondence. (c, d) Structures from the pretrained model. (e, f) Structures after RL. (c, e) Best Vinardo score. (d, f) Most distant from the best-Vinardo-score structure in TM-score.



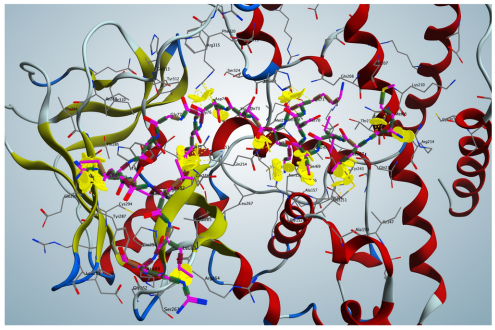
(a) Ground-truth

(b)

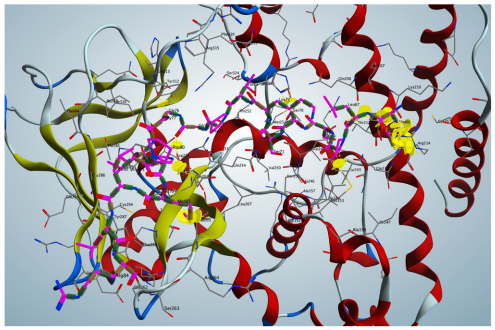
Panel	Sequence
(a)	TLKYPIEQGIVTNWDDMEK
(c)	MLISKSTSGHVGANAQLLY
(d)	MLISSKKTDIGGQRAILLR
(e)	MLPSSSKTPGGGHVPMLLR
(f)	MLISDSATGVLGGLALLLR



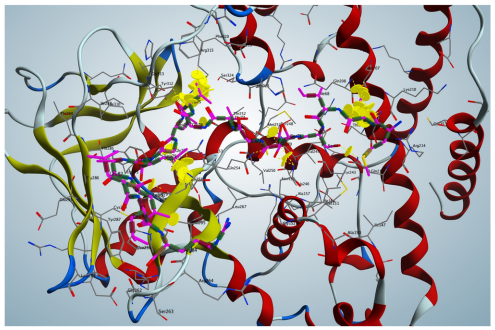
(c) Pretrained, best Vina score



(d) Pretrained, most distant



(e) RL, best Vina score



(f) RL, most distant

Figure 10: Structural validity of generated peptides for 6V63. Ligands are shown in magenta. Torsional strain of rotatable bonds is assessed using the Ligand Torsions tool in MOE, where cylinders along rotatable bonds are color-coded by geometric quality (green: good; orange: warning; red: bad). Protein–ligand steric clashes are indicated by yellow dashed lines; yellow disks denote the clash energy magnitude (threshold: 10 kcal/mol). (a) Ground-truth peptide. (b) Panel-to-sequence correspondence. (c, d) Structures from the pretrained model. (e, f) Structures after RL. (c, e) Best Vinardo score. (d, f) Most distant from the best-Vinardo-score structure in TM-score.

Algorithm 3 Model forward pass (f_θ). h_0, z_0 are pre-computed from $\mathcal{C}^{\text{rec}}$ once before generation (see Algorithm 1); peptide token slots are blank.

Require: Peptide state $\mathcal{C}_t^{\text{pep}} = (s_t, R_t, x_t, \chi_t^{20\times5})$; time t; node repr. h_0; edge repr. z_0
Ensure: Predicted $t{=}1$ state $(\hat{s}_{1,\theta},\ \hat{R}_1,\ \hat{x}_1,\ \hat{\chi}_{1,\theta}^{20\times5})$

$e_t \leftarrow \textsc{TimeEmbed}(t)$ ▷ sinusoidal + 2-layer MLP
$(h_{\text{sc}},\ h_{\text{aa}}) \leftarrow \textsc{SideChainEncode}(\chi_t, s_t)$ ▷ Alg. 5
$h_{\text{time}} \leftarrow \text{Linear}(\text{SiLU}(e_t))$
$h \leftarrow \text{MLP}\big([h_0;\ h_{\text{sc}};\ h_{\text{aa}};\ h_{\text{time}}]\big)$ ▷ fuse with pre-computed node repr.
$z \leftarrow z_0$
$R,\ x \leftarrow R_t,\ x_t$
for $\ell = 1, \ldots, L$ **do**
  $(R, x, h, z) \leftarrow \textsc{FoldIteration}_\ell(e_t, R, x, h, z)$ ▷ Alg. 4
end for
$\hat{s}_{1,\theta} \leftarrow \textsc{SeqNet}(h)$ ▷ residual MLPs + Linear; $\in \mathbb{R}^{N\times20}$
$\hat{\chi}_{1,\theta} \leftarrow \textsc{AngleNet}(h)$ ▷ residual MLPs + Linear; $\in \mathbb{R}^{N\times20\times5}$
return $(\hat{s}_{1,\theta},\ R,\ x,\ \hat{\chi}_{1,\theta})$

Algorithm 4 FoldIteration block. $\text{AdaLN}_0(h, e_t, f)$ denotes AdaLN-Zero conditioning (Peebles & Xie, 2023): $(\gamma, \beta, g) \leftarrow \text{Linear}_0(\text{SiLU}(\text{Linear}(e_t)))$ (subscript $_0$: zero-initialized output); return $h + g \odot f\big((1{+}\gamma) \odot \text{LN}(h) + \beta\big)$.

Require: Time embedding e_t; rotations R; translations x; node repr. h; edge repr. z
Ensure: Updated (R, x, h, z)

$h \leftarrow \text{AdaLN}_0\big(h,\ e_t,\ \text{IPA}(\cdot, z, R, x)\big)$ ▷ Invariant Point Attention
for each Transformer sub-layer **do**
  $h \leftarrow \text{AdaLN}_0\big(h,\ e_t,\ \textsc{SelfAttn}(\cdot)\big)$ ▷ multi-head self-attention
  $h \leftarrow \text{AdaLN}_0\big(h,\ e_t,\ \text{FFN}(\cdot)\big)$ ▷ feed-forward network
end for
$h \leftarrow \text{AdaLN}_0\big(h,\ e_t,\ \textsc{Transition}(\cdot)\big)$ ▷ 3-layer MLP with SiLU
$(R, x) \leftarrow \textsc{BackboneUpdate}(h, R, x)$ ▷ peptide residues only; receptor unchanged (Jumper et al., 2021)
$z \leftarrow \text{AdaLN}_0\big(z,\ e_t,\ \textsc{EdgeTransition}(\cdot, h)\big)$ ▷ omitted in the last block
return (R, x, h, z)

at the start of training AdaLN-Zero acts as the identity, and time dependence is acquired gradually through learning.

F.2 Side-Chain Feature Construction

For each residue j and amino acid type a, side-chain features are constructed from the predicted torsion angles $\chi_{1,\ldots,5}^{j,a}$ via two pathways.

Heavy-atom coordinates in the backbone frame. The torsion angles are used to reconstruct the 3D coordinates of side-chain heavy atoms in the backbone frame (C_α as origin, C direction as the x-axis). These coordinates are mapped to embedding vectors through an MLP with amino-acid-specific weights (GroupLinear). This representation allows the model to directly leverage side-chain geometry as features while maintaining the $\text{SE}(3)$ invariance of the IPA module.

Sinusoidal angular embedding. Following the original PepFlow, sinusoidal embeddings at multiple frequencies are applied to the torsion angles. Angles absent for a given amino acid type are zeroed out via an existence mask before passing through an MLP.

The outputs of both pathways are summed and then aggregated into a single feature vector per residue by weighting with the amino acid probabilities $p_t(a)$, as specified in Eq. equation 1.

Algorithm 5 SIDECHAINENCODE. GROUPMLP$_a$: MLP with amino-acid-specific weights. M_a^{atom}, M_a^{tor}: per-type validity masks for side-chain atoms and torsion angles.

Require: Torsion angles $\chi \in \mathbb{R}^{N\times 20\times 5}$; sequence logits $s \in \mathbb{R}^{N\times 20}$
Ensure: Side-chain features h_{sc}; amino acid features h_{aa} $(\in \mathbb{R}^{N\times D})$

$p \leftarrow \text{softmax}(s)$ ▷ amino acid probs. $p_t(a)$
for $a = 1, \ldots, 20$ **do**
  $\text{coords}_a \leftarrow \text{RECONSTRUCTATOMS}(\chi_{:,a,:})$ ▷ torsion → 3D coords in backbone frame
  $h_a^{\text{coord}} \leftarrow \text{GROUPMLP}_a\big(\text{flatten}(\text{coords}_a \odot M_a^{\text{atom}})\big)$
  $h_a^{\text{ang}} \leftarrow \text{GROUPMLP}_a\big(\text{SINCOSEMBED}(\chi_{:,a,:}) \odot M_a^{\text{tor}}\big)$ ▷ sinusoidal frequencies
end for
$h_{\text{sc}} \leftarrow \sum_{a=1}^{20} p(a) \cdot (h_a^{\text{coord}} + h_a^{\text{ang}})$ ▷ Eq. equation 1
$h_{\text{aa}} \leftarrow \text{Linear}(p)$
return $(h_{\text{sc}},\ h_{\text{aa}})$

F.3 TORSION ANGLE HANDLING IN PRETRAINING

Our model predicts 20×5 torsion angles per residue, but the training data contains only one known amino acid type per residue, leaving no ground truth for the remaining 19 types. We handle this as follows.

During pretraining, when constructing the target torsion angles at $t = 1$, the slot for the ground-truth amino acid type a^* is filled with the true torsion angles, while slots for $a \neq a^*$ are filled with the same values as the initial noise at $t = 0$. This makes the velocity for non-ground-truth types $\chi_1 - \chi_0 = 0$ (zero velocity). The loss is computed only over the torsion angle terms corresponding to a^*; terms for $a \neq a^*$ are masked and do not contribute to the gradient.

At generation time, after integrating to $t = 1$, the amino acid sequence is determined from the predicted logits, and the torsion angles of the selected type are used as the final side-chain structure. Because pretraining assigns zero velocity to non-ground-truth types, the model learns to produce meaningful velocities only for the correct amino acid type. During RL fine-tuning, reward-based feedback updates the policy for all amino acid types, including those that were masked during pretraining.

F.4 ROSETTA VALIDATION

As an independent validation metric, we evaluate binding free energies using the Rosetta all-atom energy function (Alford et al., 2017; Chaudhury et al., 2010), which was not used during training. Since our reward function evaluates generated structures as-is without relaxation or optimization, we adopt the unrelaxed binding free energy ΔG_{bind} as the primary validation metric for consistency. ΔG_{bind} is computed by subtracting the energies of the isolated receptor and peptide from the energy of the receptor–peptide complex (lower is better).

Table 9 reports the median ΔG_{bind} across all checkpoints. ΔG_{bind} **improves with RL fine-tuning on all targets.** On 1SSH it drops from $1691 \rightarrow 17$ kcal/mol, on 2R02 from $1431 \rightarrow 206$, on 5UZW from $1912 \rightarrow 376$, and even on 6V63 from $7463 \rightarrow 5480$. These improvements are directionally consistent with the Vinardo reward, indicating that the learned reward improvement transfers to an independent scoring function when structures are evaluated without relaxation.

Sample-level correlations. Restricting to clash-free samples, the Spearman rank correlation between the Vinardo reward and ΔG_{bind} is $\rho = -0.88$ ($p < 10^{-16}$, $n = 51$) on 1SSH, $\rho = -0.94$ ($p < 10^{-20}$, $n = 45$) on 2R02, and $\rho = -0.83$ ($p < 10^{-10}$, $n = 39$) on 5UZW — very strong negative correlations (higher Vinardo corresponds to lower ΔG_{bind}). Thus the Vinardo score serves as a good predictor of the unrelaxed binding free energy.

Comparison with $\Delta G_{\text{bind}}^{\text{relaxed}}$. The rightmost column of Table 9 also reports $\Delta G_{\text{bind}}^{\text{relaxed}}$ (after fast relax) for reference. Improvement under the relaxed metric is limited: on 1SSH ($-18.20 \rightarrow -12.79$) and 2R02 ($-13.51 \rightarrow -10.71$) the relaxed binding energy actually worsens. The sample-level Spearman correlation between the Vinardo reward and $\Delta G_{\text{bind}}^{\text{relaxed}}$ is not statistically significant for

Table 9: Median Rosetta ΔG_{bind} (unrelaxed, kcal/mol; lower is better). All targets improve with RL fine-tuning. $\Delta G_{\mathrm{bind}}^{\mathrm{relaxed}}$ (after fast relax) is shown for reference.

Checkpoint	$\Delta G_{\mathrm{bind}} \downarrow$	$\Delta G_{\mathrm{bind}}^{\mathrm{relaxed}} \downarrow$	**BSR**
1SSH Pretrained	1691	-18.20	0.976
3,000	1408	-18.64	0.959
6,000	799	-16.31	0.942
9,000	247	-16.04	0.857
12,000	225	-14.82	0.795
15,000	55	-15.32	0.746
18,000	**17**	-12.79	0.717
1SSH PretrainedCE	190	-14.33	0.833
CE 3,000	68	-16.54	0.843
CE 6,000	130	-16.43	0.836
2R02 Pretrained	1431	-13.51	0.718
3,000	838	-12.26	0.704
6,000	648	-12.00	0.677
9,000	372	-10.14	0.656
12,000	**206**	-10.71	0.628
5UZW Pretrained	1912	-0.15	0.843
3,000	979	-4.20	0.845
6,000	840	-5.19	0.841
8,000	636	-3.70	0.832
11,000	478	-6.54	0.818
14,000	**376**	-5.38	0.820
6V63 Pretrained	7463	-9.91	0.758
2,000	5939	-19.74	0.756
4,000	**5480**	-18.17	0.766

any target ($|\rho| \leq 0.24$, $p > 0.08$), in stark contrast with the strong correlations observed for the unrelaxed ΔG_{bind} ($|\rho| > 0.83$).

This discrepancy reflects a limitation inherent to our reward design: since the reward function evaluates generated structures without relaxation, it aligns well with unrelaxed evaluation metrics but not with relaxed ones. In practice, relaxed or optimized structures are often used for downstream applications, so improving performance on relaxed structures would require redesigning the reward to operate on relaxed conformations. However, because locally distinct generated structures may converge to the same local optimum upon relaxation, the reward signal can vanish—an issue that compounds the computational cost of relaxation within the training loop. We therefore did not pursue this direction in the present work.