NONPARAMETRIC BIOLOGIC DESIGN VIA DISCRETE DRIFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent algorithms for sequence-based therapeutic design use diffusion and flow models as statistical priors for optimization in combinatorial spaces. Adapting these priors to new reference datasets can require additional training. We explore whether a frozen representation encoder and reference data can support such adaptation without fitting a generative model. We introduce **NimblDrift**, a *discrete* feature-space drifting algorithm for generating and optimizing populations of biologic candidates. At each step, we select local edits that reduce a discrepancy between the candidate and reference populations in feature space. On three small-molecule targets, the best-scoring drift configurations exceeded greedy-predictor and random-walk baselines in predicted binding. With **GreedyDrift**, we also improved predicted peptide affinity while retaining reference-like distributions of secondary properties. Allele-specific peptide generation further tests the functional information accessible through representation geometry. For practical design, NimblDrift supports modular search with frozen encoders and replaceable reference populations. For representation research, these experiments offer a direct way to compare encoders through their effects on discrete generation.

1 Introduction

Peptides and small molecules occupy combinatorial *design spaces* where local edits can alter binding, solubility, and other properties relevant to therapeutic development. Recent design methods use discrete diffusion and flow models as statistical priors over these spaces (Stärk et al., 2024; Tang et al., 2025; Chen et al., 2025; De Santi et al., 2026). Guidance can incorporate property objectives into the learned dynamics, although adapting the underlying prior to a new reference dataset can require retraining or fine-tuning. Pretrained protein and molecular encoders offer a complementary source of information by mapping candidates into a *feature space* that can reflect sequence, structure, and function (Lin et al., 2023; Chithrananda et al., 2020; Ross et al., 2022; Soares et al., 2025; Feller et al., 2026). We therefore ask whether reference data and representation geometry can support population generation directly.

Can we pair a representation encoder with reference data to form an effective and generalizable statistical prior for sequence-based biologic design?

We call a method effective when generated candidates reflect properties of the reference data, and assess generality across tasks and reference distributions. Focusing on a simple instantiation, we introduce **NimblDrift**, a **N**onparametr**i**c **m**ethod for **b**io**l**ogic design via feature-space **Drift**ing (Deng et al., 2026). We iteratively edit population members to reduce a discrepancy from a reference population in a frozen encoder's feature space. The same discrepancy can regularize reward optimization. Here, the reference-induced prior specifies preferences for candidate populations through their feature distributions. Generation proceeds by local search using this reference-based preference. Our contributions span the update rule, its dependence on neighborhood structure, and biological design applications.

1. **Discrete Drifting and Reward Control.** We adapt feature-space drifting (Deng et al., 2026) to discrete spaces through sequential population updates and introduce **GreedyDrift** for reward optimization with a reference-based regularizer.

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.

2. **Analysis via Small-World Networks.** We connect singleton drift to greedy graph routing and use synthetic networks (Kleinberg, 2000) to examine convergence. This analysis motivates crossover neighborhoods for small-molecule generation.
3. **Evaluation on Biological Design Tasks.** We evaluate predicted small-molecule binding, peptide property retention during binding optimization, and allele-specific MHC presentation. A sentence-level example first illustrates the update in an interpretable setting.

2 Background

2.1 Distribution Matching in Feature Space

Generative models learn a distribution q_θ, parameterized by θ, using objectives that compare it with a data distribution p_{data}. Autoregressive language models minimize token-level cross-entropy, and diffusion models commonly use denoising score matching. The data representation and training objective determine which discrepancies are penalized, with generalization dependent on the model, data, and training procedure.

Feature matching compares distributions after applying an encoder ϕ. For example, Salimans et al. (2016) matched discriminator feature statistics to stabilize GAN training. More recently, Deng et al. (2026) trained image generators using attraction and repulsion in pretrained feature spaces. Particular drift fields have connections to Wasserstein gradient flows, with the interpretation depending on the kernel and normalization (Gretton et al., 2026). We retain the term "drift" for population updates motivated by this feature-space matching approach.

2.2 Sequence-based Models for Biologic Design

Peptides and small molecules admit discrete representations as amino-acid sequences and molecular graphs or strings. Even length-12 peptides over the 20 canonical amino acids span $20^{12} \approx 4 \times 10^{15}$ sequences, motivating computational search before experimental screening (Gao et al., 2022; Trabucco et al., 2022).

Sequence-based models support three overlapping roles in this search. Property predictors $\hat{f}_i : \mathcal{X} \to \mathbb{R}$ estimate a candidate's properties in a design space $\mathcal{X}$. Generative priors approximate a data distribution from which candidates can be sampled. Encoders $\phi : \mathcal{X} \to \mathbb{R}^d$ map candidates to d-dimensional representations used for downstream prediction or similarity comparisons. We use frozen encoders to guide discrete population updates without fitting a decoder or a new generative model.

3 Method

3.1 Discrete Drift

Let $\mathcal{X}$ be a combinatorial design space, $\mathcal{N}(x) \subseteq \mathcal{X}$ the candidate neighborhood of x, and $\phi : \mathcal{X} \to \mathbb{R}^d$ a frozen encoder. We include the unchanged candidate through $\mathcal{N}^+(x) = \mathcal{N}(x) \cup \{x\}$. At iteration t, the ordered population $\mathcal{Q}_t = (x_t^{(1)}, \ldots, x_t^{(M)})$ contains M candidates, with duplicates permitted. For N reference candidates $\mathcal{P}_{\text{ref}} = (y_1, \ldots, y_N)$, the empirical distributions are

$$q_t = \frac{1}{M} \sum_{i=1}^{M} \delta_{x_t^{(i)}}, \qquad p_{\text{ref}} = \sum_{j=1}^{N} w_j \delta_{y_j}, \qquad w_j \geq 0, \quad \sum_{j=1}^{N} w_j = 1. \tag{1}$$

Each Dirac mass δ_x places unit mass at candidate x, and the reference weights default to $w_j = 1/N$. The pushforward $\phi_\# q = \sum_x q(x) \delta_{\phi(x)}$ maps a sequence distribution to its feature distribution.

For neural embeddings, we smooth both feature distributions with the Gaussian density $k_\tau(z) = (2\pi\tau^2)^{-d/2} \exp[-\|z\|_2^2/(2\tau^2)]$, where $\tau > 0$ is the bandwidth. For a candidate population $q = M^{-1} \sum_i \delta_{x^{(i)}}$, let $z_i = \phi(x^{(i)})$ and $r_j = \phi(y_j)$. The smoothed densities are

$$\widehat{q}_\tau(z) = \frac{1}{M} \sum_{i=1}^{M} k_\tau(z - z_i), \qquad \widehat{p}_\tau(z) = \sum_{j=1}^{N} w_j k_\tau(z - r_j). \tag{2}$$

We evaluate their log-density ratio at the candidate embeddings to obtain the empirical reverse-KL surrogate

$$\mathcal{D}_t(q, p_{\text{ref}}) = \frac{1}{M} \sum_{i=1}^{M} \log \frac{\widehat{q}_{\tau_t}(z_i)}{\widehat{p}_{\tau_t}(z_i)}. \tag{3}$$

This score compares candidate density with reference density at each candidate. Its averaging measure is $\phi_{\#}q$, so it differs from the exact KL integral under $\widehat{q}_{\tau_t}$. Appendix A1.1 details this distinction and the population-dependent bandwidth, which remains fixed during each sweep. For Morgan fingerprints (FP) (Rogers & Hahn, 2010), we use maximum mean discrepancy (MMD) with a Tanimoto similarity kernel.

We minimize $\mathcal{D}_t$ by sequential coordinate updates. Given the current population $q = M^{-1} \sum_k \delta_{x^{(k)}}$, define $\mathcal{B}_i(q) = \{q + M^{-1}(\delta_u - \delta_{x^{(i)}}) : u \in \mathcal{N}^+(x^{(i)})\}$. Starting from $q_t^{(0)} = q_t$, the NimblDrift sweep is

$$q_t^{(i)} \in \operatorname*{arg\,min}_{q \in \mathcal{B}_i(q_t^{(i-1)})} \mathcal{D}_t(q, p_{\text{ref}}), \quad i = 1, \ldots, M, \qquad q_{t+1} = q_t^{(M)}. \tag{4}$$

Each coordinate update selects the lowest-scoring replacement using the already updated population. Including the current candidate ensures nonincrease within a sweep, retaining it on ties. The joint neighborhood $\mathcal{B}(q_t) = \{M^{-1} \sum_i \delta_{u_i} : u_i \in \mathcal{N}^+(x_t^{(i)})\}$ contains all simultaneous replacements, whose joint minimum is generally more expensive to compute. We return the population after the prescribed number of sweeps.

3.2 Incorporating Reward Optimization into the Discrete Drift

For a scalar reward $R : \mathcal{X} \to \mathbb{R}$ and regularization weight $\lambda \geq 0$, **GreedyDrift** maximizes mean population reward penalized by reference discrepancy,

$$\mathcal{J}_t(q) = \mathbb{E}_{x \sim q}[R(x)] - \lambda \mathcal{D}_t(q, p_{\text{ref}}), \qquad q_t^{(i)} \in \operatorname*{arg\,max}_{q \in \mathcal{B}_i(q_t^{(i-1)})} \mathcal{J}_t(q). \tag{5}$$

The expectation is the mean reward over the M candidates. Larger λ increases the penalty for departures from the reference in feature space, and $\lambda = 0$ recovers unregularized greedy search. In implementation, we use a ranking of candidates to normalize the numerical scales of the reward and discrepency term (see Appendix A1.1). The same sequential sweep applies, with $q_{t+1} = q_t^{(M)}$. More generally, a population functional $\mathcal{F} : \Delta(\mathcal{X}) \to \mathbb{R}$ can replace the mean reward, where $\Delta(\mathcal{X})$ denotes probability distributions on $\mathcal{X}$. A lower-tail conditional value-at-risk (CVaR) objective, for example, would emphasize the worst-scoring candidates (Chen et al., 2026).

4 Empirical Analysis

4.1 Small-World Networks and Greedy Routing

NimblDrift dynamics depend on both the design space $\mathcal{X}$ and its neighborhood structure $\mathcal{N}$. To examine this dependence, consider the neighborhood graph $G = (V, E)$ with vertices $V = \mathcal{X}$ and edges $E = \{(u, v) : v \in \mathcal{N}(u)\}$. For singleton populations $q_t = \delta_{x_t}$ and $p_{\text{ref}} = \delta_y$, the Gaussian score in Equation (3) reduces to $\|\phi(x) - \phi(y)\|_2^2/(2\tau_t^2)$. Thus, the update becomes greedy routing by feature distance,

$$x_{t+1} \in \operatorname*{arg\,min}_{x \in \mathcal{N}^+(x_t)} \|\phi(x) - \phi(y)\|_2. \tag{6}$$

This rule selects the neighbor closest to the target embedding. We use synthetic small-world graphs (Kleinberg, 2000) to examine how long-range edges affect such updates.

Kleinberg's construction augments lattice edges with randomly sampled long-range connections. In two dimensions, sampling shortcuts with probability proportional to inverse squared lattice distance permits greedy routing in an expected $O((\log n)^2)$ steps on an $n \times n$ lattice (Kleinberg, 2000). Our synthetic experiments illustrate the effect of these shortcuts on population drift (Figure 1). The routing guarantee concerns Kleinberg's graph and distance assumptions, and does not establish a convergence rate for molecular search.

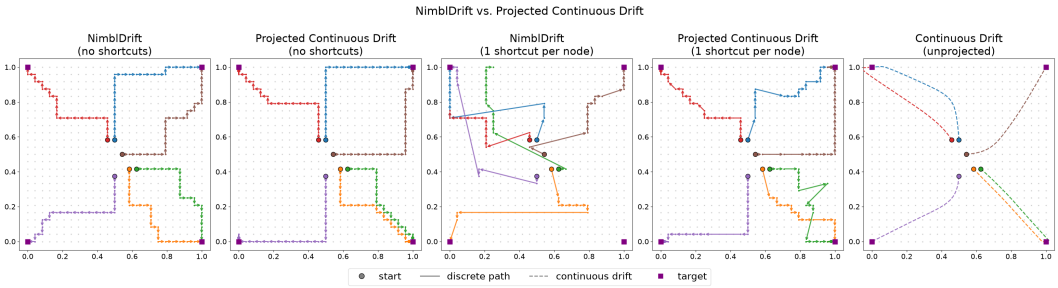

Figure 1: **NimblDrift and projected continuous drift in a Kleinberg network.** The first two panels compare trajectories without long-range edges, and the next two include these edges. Direct discrepancy minimization permits shortcuts that reduce the population objective. Projected updates favor neighbors close to the continuous displacement. The final panel overlays a continuous trajectory. The underlying graph appears in Appendix Figure A1.

4.2 A Comparison to Projecting the Continuous Drift

A direct adaptation of feature-space drifting (Eq. 13) from Deng et al. (2026) to discrete candidates offers an alternative algorithm to move candidate populations. Given a drift displacement $\mathbf{V}_t : \mathbb{R}^d \to \mathbb{R}^d$, it selects a neighboring population nearest to the displaced embeddings,

$$q_{t+1} \in \arg\min_{q \in \mathcal{B}(q_t)} \sum_{i=1}^{M} \left\| \phi(x^{(i)}) - \left[\phi(x_t^{(i)}) + \mathbf{V}_t(\phi(x_t^{(i)})) \right] \right\|_2 . \tag{7}$$

Here $x^{(i)}$ is the ith candidate in q, and the displacement scale is included in $\mathbf{V}_t$. For the gaussian kernel, $\mathbf{V}_t$ estimates the reverse-KL Wasserstein gradient-flow direction of the drift population towards the reference (Gretton et al., 2026). Projection favors edits near the continuous update. Direct discrepancy minimization can select a shortcut with lower population discrepancy even when its endpoint is far from that update.

In the illustrated network, 2/6 NimblDrift trajectories ended at the same mode as their continuous counterparts, compared with 5/6 projected-drift trajectories (Figure 1). Across random shortcut realizations, the reported mean convergence times were 10.67 and 14.17 steps, respectively. We observed a similar tradeoff between discrepancy reduction and agreement with continuous trajectories in the peptide experiment in Appendix A2.2.

4.3 Designing Molecular Neighborhoods with Long-Range Edits

The small-world experiments motivate examining neighborhood connectivity alongside the update rule. For molecular design, the neighborhood graph induced by $(\mathcal{X}, \mathcal{N})$ depends on which chemical edits are allowed. Single molecular-graph edits[1] led to slow convergence in our experiments. Motivated by the graph-based genetic algorithm Graph GA (Jensen, 2019; Gao et al., 2022), we augmented these neighborhoods with crossover proposals from K partners sampled uniformly without replacement from a buffer B,

$$\mathcal{N}_{B\text{-crossover}}(x) = \bigcup_{k=1}^{K} \mathcal{C}(x, a_k), \qquad \{a_k\}_{k=1}^{K} \sim \text{UniformSubsets}_K(B). \tag{8}$$

Here $\mathcal{C}(x, a_k)$ contains valid, deduplicated offspring from both parental orders, including the mutation step in Appendix A1.2. The buffer is either a BindingDB corpus ($\mathcal{N}_{\text{BDB-crossover}}$) or the current population ($\mathcal{N}_{\text{GA-crossover}}$). These proposals can make several molecular-graph edits in one step, imitating distribution of Kleinberg's model.

Augmenting $\mathcal{N}_{\text{graph-edit}}$ with crossover proposals reduced discrepancy more rapidly across the tested feature spaces. With Morgan fingerprints, within-population crossovers reached the reference self-MMD^2 benchmark in 45 steps, compared with 92 for BindingDB crossovers and more than 100 for graph edits alone (Table 1). Appendix A2.3 describes this comparison.

[1] A molecular graph represents atoms and bonds. The neighborhood graph connects entire molecules through allowed edits.

Table 1: Final convergence of small-molecule drift populations towards a reference population of antibiotics under three neighborhood constructions. $K = 10$ for all crossovers.

Neighborhood	Min. $\text{MMD}^2 \downarrow$	Steps to self-MMD^2 floor (0.0065) $\downarrow$
$\mathcal{N}_{\text{graph-edit}}$	0.0128	> 100
$\mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{BDB-crossover}}$ (B = BindingDB)	0.0062	92
$\mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{GA-crossover}}$ ($B = q_t$)	**0.0049**	**45**

5 Applications to Design Problems

We evaluated NimblDrift across several modalities and feature spaces, starting with sentences as an interpretable example. The biological applications tested small-molecule binding scores, peptide property retention during reward optimization, and allele-specific MHC presentation scores. Appendix A3 contains additional experimental details.

5.1 Drifting Sentences

Background. Sentences offer an interpretable example of editing discrete sequences under a reference-based objective. We applied NimblDrift to negative-sentiment sentences using a positive-sentiment reference population.

Experimental Setup. We used language models to generate the initial and reference populations and to propose neighboring sentences (Table 2).

Table 2: Experimental setup for the sentence sentiment drift.

Component	Choice
Initial Population	100 sentences about tragic events in Shakespeare, generated by Claude Sonnet 5
Reference Population	100 sentences about uplifting events from the news generated by Claude Sonnet 5
Embedding / Discrepancy	Gemma 300M / Gaussian log-density-ratio surrogate
Neighborhood	Qwen2.5-14B proposes 10 similar sentences per population member as a semantic neighborhood
Reward Signal	None
Evaluation Metrics	VADER sentiment score
Baseline	None
Timesteps	5

Results. The terminal population contained less negative descriptions, with some loss of event-specific detail. For example, "Ophelia descends into madness and drowns after her father is killed by the man she loves." changed to "Her heart aches as she reflects on the past with her now deceased loved one." and finally "Wistfully, she reminisces about the now-gone days filled with love and longing". The median VADER sentiment score increased from -0.73 to 0.50, consistent with this shift in tone.

5.2 Small-Molecule Binder Generation

Background. Small-molecule binding datasets from BindingDB (Liu et al., 2007; 2025) and PubChem (Kim et al., 2023) support both reference-based design and affinity prediction. We asked whether moving a molecular population toward experimentally characterized binders in feature space would increase independently computed binding scores.

Experimental Setup. We generated candidates for targets represented by PDB structures 2VT4, 6CM4, and 5SDV. These targets were selected from Chen et al. (2026) because literature binders or target-specific BindingDB measurements were available (Table 3).

Table 3: Experimental setup for small-molecule binder generation.

Component	Choice
Initial Population	100 random BindingDB ligands
Reference Population	Fewer than 10 literature binders ("known binders"), with an expanded reference adding 200 ligands from target-specific BindingDB assays
Embedding / Discrepancy	{SMI-TED$_{289M}$, Molformer, Morgan FP} with the discrepancies in Section 3.1
Neighborhood	$\{\mathcal{N}_{\text{graph-edit}}, \mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{BDB-crossover}}, \mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{GA-crossover}}\}$, $K = 10$ crossovers
Reward Signal	None
Evaluation Metrics	Binding likelihood (p_{bind}) to the specified target as scored by Boltz-2 (Passaro et al., 2025)
Baseline	FlashBind (Jiang et al., 2026) greedy optimization and random walk, both in $\mathcal{N}_{\text{BDB-crossover}}$ neighborhoods
Timesteps	40

We also tested reference weights proportional to each reference ligand's Boltz-2 score (Appendix A3.1). This variant uses predictor information to construct the reference distribution, with no direct reward on newly proposed molecules. Because Boltz-2 also scores the final candidates, evaluation of this variant is not independent of reference weighting.

Results. Over 40 steps, the strongest configurations exceeded the random-walk and FlashBind baselines on all three targets (Table 4). With the weighted reference set and within-population crossovers, SMI-TED$_{289M}$ achieved mean p_{bind} values of 0.544 on 2VT4 and 0.812 on 6CM4. Molformer attained 0.423 on 5SDV, though SMI-TED$_{289M}$ was nearby at 0.419. This result shows that NimblDrift reflects the improved general performance of SMI-TED$_{289M}$ over Molformer. At this population size we note that Morgan Fingerprints performed just above the random walk baseline, though we found that performance may be improved by increasing the size of the drift population. We study this and the tradeoff between bind scores and sequence novelty in Appendix A3.1. The trajectories in Figure 2 report the 75th percentile of p_{bind} for the three representations and the random-walk baseline.

Table 4: Final mean p_{bind} (Boltz-2) at the last drift/search step. First row provides a baseline score (random walk in the crossover neighborhood). We only score subsets of termianl drift populations due to computational expense of Boltz-2 fold prediction. Drift rows (Morgan/Molformer/SMI-TED$_{289M}$) pool two independently-reseeded drifts with 10 molecules each to compute a bootstrap mean; Random Walk and FlashBind Greedy use a bootstrap mean of 10 independent runs, not reseeded. Highest score per target in **bold**.

Method	Configuration	2VT4	5SDV	6CM4
Random Walk	–	0.197	0.191	0.156
FlashBind Greedy	–	0.187	0.285	0.094
Morgan FP Drift	known binders	0.256	0.185	0.146
	+ BindingDB unweighted + $\mathcal{N}_{\text{GA-crossover}}$	0.355	0.291	0.439
Molformer Drift	known binders	0.290	0.233	0.227
	+ BindingDB weighted + $\mathcal{N}_{\text{GA-crossover}}$	0.336	**0.423**	0.721
SMI-TED$_{289M}$ Drift	known binders	0.454	0.237	0.356
	+ BindingDB weighted + $\mathcal{N}_{\text{GA-crossover}}$	**0.544**	0.419	**0.812**

5.3 Therapeutic Viability in Peptide Binding Optimization

Background. Peptide development depends on properties beyond target binding, including solubility and hemolytic activity. With PeptiVerse, Zhang et al. (2026) assembled predictors for these properties and binding affinity. We asked whether GreedyDrift could improve predicted affinity while retaining property distributions similar to a natural-peptide reference, using affinity as the only reward.

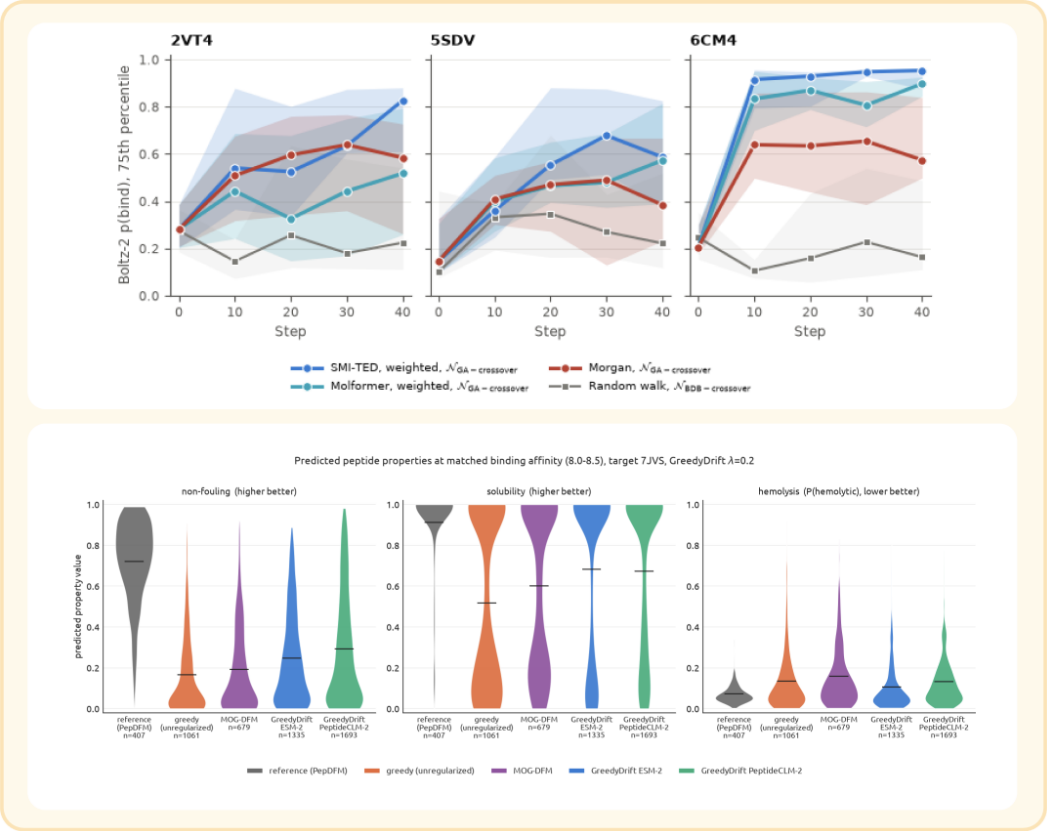

Figure 2: **NimblDrift in therapeutic design spaces.** *Top.* The 75th percentile of Boltz-2 p_{bind} during small-molecule generation for three targets. SMI-TED and Molformer runs used weighted references, and the Morgan-fingerprint run used an unweighted reference. Error bars are 95% confidence bands via bootstrapping 10 terminal members over three seends. *Bottom.* Predicted peptide property distributions for target 7JVS among candidates with binding-affinity scores in $[8.0, 8.5]$. PeptideCLM-2 regularization retained favorable solubility, non-fouling, and hemolysis profiles at comparable affinity scores. These quantities are computational predictions.

Experimental Setup. We optimized predicted binding to the target represented by PDB 7JVS while regularizing toward naturally occurring 12-mers. Secondary-property predictors were used after optimization to evaluate departures from the reference distributions (Table 5).

Table 5: Experimental setup for therapeutic regularization in peptide binding optimization.

Component	Choice
Initial Population	407 uniformly random 12-mers
Reference Population	407 naturally occurring 12-mers from the PepDFM (Chen et al., 2025) training corpus
Embedding / Discrepancy	{ESM-2 35M, PeptideCLM-2} mean-pooled embedding (Lin et al., 2023; Feller et al., 2026)
Neighborhood	$\mathcal{N}_{\text{single-substitution}}$
Reward Signal	PeptiVerse binding-affinity predictor (Zhang et al., 2026)
Evaluation Metrics	PeptiVerse non-fouling, solubility, and hemolysis predictors
Baseline	Unregularized greedy optimization, MOG-DFM (Chen et al., 2025)
Timesteps	15

Results. Increasing λ reduced the shift in secondary-property distributions at the cost of lower predicted affinity (Table 6). At $\lambda = 0.2$, mean affinity increased from 6.281 initially to 7.969, and solubility earth-mover distance (EMD) was 0.280, compared with 0.670 under unregularized search.

Does this regularization also improve secondary properties at comparable affinity? When $\lambda = 0.2$, all four methods generated candidates with affinity scores in $[8.0, 8.5]$, enabling a score-matched comparison (Figure 2). PeptideCLM-2 candidates generated binders in the favorable regions of all properties. ESM-2 candidates performed similarly, though fairing slighly worse with fewer no-fouling and more non-soluble binders. These comparisons support improved property retention within the evaluated affinity band.

Table 6: Earth-mover distance (EMD) of each secondary property's distribution to the natural-peptide reference population, and mean predicted binding affinity, at the final GreedyDrift step. Results are for PeptideCLM-2 regularization.

	EMD to natural population (↓)			
Population	Non-fouling	Solubility	Non-hemolytic	Binding affinity (↑)
Initial (random 12-mers)	0.101	0.010	0.008	6.281
$\lambda = 0.0$ (Greedy)	0.602	0.670	0.122	8.651
$\lambda = 0.1$	0.442	0.411	0.102	8.294
$\lambda = 0.2$	0.387	0.280	0.089	7.969
$\lambda = 0.5$	0.327	0.139	0.033	7.431
$\lambda = 1.0$	0.260	0.022	0.019	7.052

5.4 Allele-Specific Peptide Generation for MHC Presentation

Background. Major histocompatibility complex (MHC) proteins form peptide-MHC (pMHC) complexes that present self and foreign peptides to T cells. Class I presentation predominantly involves intracellularly derived peptides recognized by $CD8^+$ T cells, and class II presentation predominantly involves endosomal processing and $CD4^+$ T cells. Eluted ligands (ELs) are peptides experimentally recovered from MHC complexes, commonly by mass spectrometry. The Immune Epitope Database (IEDB) catalogs such measurements (Vita et al., 2025), and NetMHCpan-family models predict presentation from binding and EL data (Reynisson et al., 2020). These data allow us to test whether feature-space matching preserves allele-associated sequence preferences.

Experimental Setup. We generated candidates toward allele-specific EL reference populations for class I alleles {A*01:01, A*02:01, A*03:01, A*24:02, B*07:02} and class II alleles {DRB1*01:01, DRB1*04:01, DRB1*07:01, DRB1*15:01, DRB3*02:02}. Table 7 specifies the setup. We evaluated terminal populations across all alleles of the corresponding class using EL percentile rank, for which lower values indicate stronger predicted presentation.

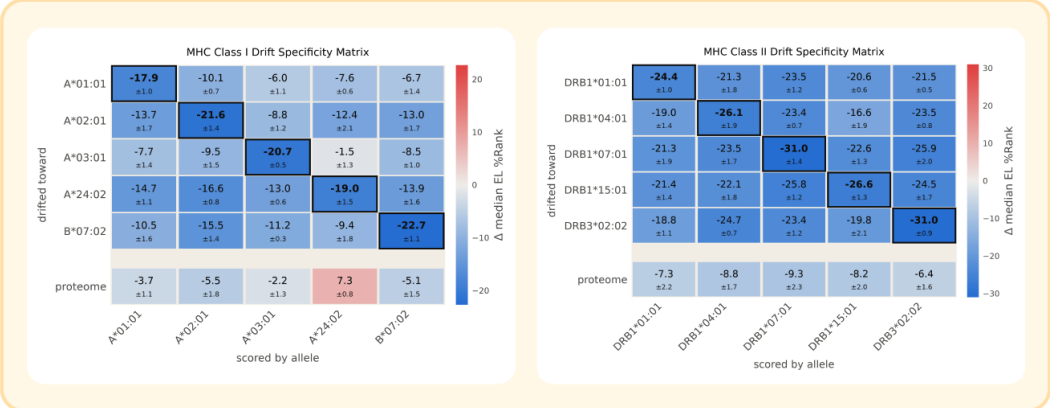

Figure 3: **Allele specificity of generated peptide candidates.** Rows correspond to the EL reference allele or human-proteome control, and columns to the evaluation allele. Cells report changes in median EL percentile rank for class I 9-mers (left) and class II 15-mers (right). Negative changes indicate stronger predicted presentation relative to initialization. Outlined diagonal cells correspond to matching reference and evaluation alleles. Error bands are computed as standard error of the mean over 6 resampled populations

Table 7: Experimental setup for eluted ligand generation in the pMHC pathway.

Component	Choice
Initial Population	200 uniformly random 9-mers (Class I) or 15-mers (Class II)
Reference Population	200 allele-specific IEDB EL peptides of length 9 (class I) or 15 (class II). Separate control drifts used human-proteome peptides of matched length
Embedding	ESM-2 35M mean-pooled embedding (Lin et al., 2023)
Neighborhood	$\mathcal{N}_{\text{single-substitution}}$
Reward Signal	None
Evaluation Metrics	Change in median EL %Rank from NetMHCpan-4.1 (class I) or NetMHCIIpan-4.0 (class II) (Reynisson et al., 2020)
Baseline	None
Timesteps	20

Results. Drift toward EL references reduced the corresponding NetMHCpan-4.1 and NetMHCIIpan-4.0 percentile ranks, with the strongest reductions generally occurring for the target allele (Figure 3). For example, drift toward A*01:01 improved A*02:01 scores less than drift toward A*02:01's own reference population. Off-diagonal improvements were more extensive for class II, indicating weaker allele discrimination in this setting. Did the generated peptides retain useful residue ordering, or only reference-like amino-acid composition? Scrambling terminal sequences worsened diagonal presentation scores (Appendix A3.3), supporting an effect of residue order beyond composition. This analysis assesses predicted presentation and does not establish T-cell immunogenicity.

6 Conclusion

In this work, we introduce **NimblDrift** for discrete population search guided by frozen representations and reference data. Across the evaluated molecular and peptide tasks, feature-space matching improved predicted binding or presentation, and GreedyDrift retained reference-like property distributions during reward optimization. These results support the central premise that a reference population and an encoder can define a useful statistical preference for discrete design without additional model training.

The method also supports comparisons of representation encoders through their effects on generation. In molecular binder generation, SMI-TED$_{289M}$ exceeded or matched Molformer on all three molecular targets. Such comparisons could help select encoders and reference datasets before training larger generative models.

We still need to validate candidate binding, presentation, and therapeutic suitability experimentally. Our predictor-based evaluations may reflect training-data overlap and correlated errors, particularly when we use the same predictor to weight references and score generated candidates. Shared information across representations can further limit evaluation independence (Huh et al., 2024). To strengthen this evidence, we should pair experimental characterization with analyses of candidate novelty and sensitivity to reference composition. Beyond these evaluations, we can investigate whether discrete drifting also supports generative-model training (Deng et al., 2026). These directions build on the flexibility demonstrated here, where we used NimblDrift to adapt biologic design across reference populations and optimization objectives using frozen representations, without retraining a generative model.

7 AI Use Statement

We used AI tools to assist with building research infrastructure and paper formating. Language models also generated sentence populations and candidate edits for the experiment in Section 5.1. The authors take responsibility for the manuscript's scientific claims, analyses, and final content.

8 Reproducibility Statement

The main text and appendix describe the population updates, discrepancy calculations, reference datasets, pretrained encoders, neighborhood operations, and evaluation procedures. Algorithm A1 provides pseudocode for the sequential update. We plan to release the complete implementation and experiment configurations with the paper.

9 Ethics Statement

This study reports computational experiments using existing datasets and pretrained models and involved no new human-participant or animal research. Predicted binding, MHC presentation, and peptide properties require experimental validation and do not establish therapeutic safety or efficacy. Experimental applications of generated candidates should undergo appropriate institutional review and biosafety assessment.

References

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. In *ICML 2025 Generative AI and Biology Workshop*, 2025. URL https://openreview.net/forum?id=8YIMLoHP9J. Spotlight.

Trevor Chen, Ariel Dai, Jason Yang, Riccardo De Santi, Daniel Khalil, Wenda Chu, Nate Gruver, Pranav Murugan, Alexander F G Goldberg, Maruan Al-Shedivat, and Yisong Yue. On the design space of discrete diffusion online adaptation for molecular optimization. *arXiv preprint arXiv:2607.02834*, 2026.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Pawel Dabrowski-Tumanski, Bartosz Topolski, Dariusz Plewczynski, and Tomasz Jetka. The geometry of activity cliffs: Representation dependence and multi-scale characterization of activity landscapes. *arXiv preprint arXiv:2605.30831*, 2026. URL https://arxiv.org/abs/2605.30831.

Riccardo De Santi, Bruce Lee, Cristian Perez Jensen, Kimon Protopapas, Sophia Tang, Cheng-Hao Liu, Pranam Chatterjee, Yisong Yue, and Andreas Krause. Active flow expansion for out-of-distribution discovery: from theory to molecules. *arXiv preprint arXiv:2606.08802*, 2026. URL https://arxiv.org/abs/2606.08802.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026. URL https://arxiv.org/abs/2602.04770.

Aaron L. Feller, Maxim Secor, Sebastian Swanson, Claus O. Wilke, and Kristine Deibler. Scaling SMILES-based chemical language models for therapeutic peptide engineering. *Journal of Chemical Information and Modeling*, 66(14):8110–8122, 2026. doi: 10.1021/acs.jcim.6c00652. URL https://doi.org/10.1021/acs.jcim.6c00652.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W Coley. Sample efficiency matters: a benchmark for practical molecular optimization. *arXiv preprint arXiv:2206.12411*, 2022.

Arthur Gretton, Li Kevin Wenliang, Alexandre Galashov, James Thornton, Valentin De Bortoli, and Arnaud Doucet. On the wasserstein gradient flow interpretation of drifting models. *arXiv preprint arXiv:2605.05118*, 2026. URL https://arxiv.org/abs/2605.05118.

Malte Grieswelle, Samuel K. R. Homberg, Philipp Janssen, Johannes Kaminski, Joana Massa, Ann-Kathrin Prinz, Stefan Georgiev, and Oliver Koch. A new benchmark for deep-learning-based affinity prediction: Solving the inter-protein scoring noise problem. *ChemRxiv*, 2026. doi: 10.26434/chemrxiv-2025-sf3cs. URL https://doi.org/10.26434/chemrxiv-2025-sf3cs.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Songlin Jiang, Yifan Chen, Aarti Krishnan, Yu Zhang, and Wengong Jin. Flashbind: Towards accurate and efficient structure-based virtual screening. *Transactions on Machine Learning Research*, 2026.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 2023. doi: 10.1093/nar/gkac956. URL https://doi.org/10.1093/nar/gkac956.

Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 163–170, 2000.

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

Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N. Jorissen, and Michael K. Gilson. BindingDB: A web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Research*, 35(D1):D198–D201, 2007. doi: 10.1093/nar/gkl999. URL https://doi.org/10.1093/nar/gkl999.

Tiqing Liu, Linda Hwang, Stephen K. Burley, Carmen I. Nitsche, Christopher Southan, W. Patrick Walters, and Michael K. Gilson. BindingDB in 2024: A FAIR knowledgebase of protein-small molecule binding data. *Nucleic Acids Research*, 53(D1):D1633–D1644, 2025. doi: 10.1093/nar/gkae1075. URL https://doi.org/10.1093/nar/gkae1075.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Birkir Reynisson, Bruno Alvarez, Sinu Paul, Bjoern Peters, and Morten Nielsen. NetMHCpan-4.1 and NetMHCIIpan-4.0: Improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Research*, 48(W1): W449–W454, 2020. doi: 10.1093/nar/gkaa379. URL https://doi.org/10.1093/nar/gkaa379.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4:1256–1264, 2022. doi: 10.1038/s42256-022-00580-7. URL https://doi.org/10.1038/s42256-022-00580-7.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Gabriele Scalia, Steven T. Rutherford, Ziqing Lu, Kerry R. Buchholz, Nicholas Skelton, Kangway Chuang, Nathaniel Diamant, Jan-Christian Hütter, Jerome-Maxim Luescher, Anh Miu, et al. Deep-learning-based virtual screening of antibacterial compounds. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02814-6. URL https://doi.org/10.1038/s41587-025-02814-6.

Eduardo Soares, Emilio Vital Brazil, Victor Shirasuna, Dmitry Zubarev, Renato Cerqueira, and Kristin Schmidt. An open-source family of large encoder-decoder foundation models for chemistry. *Communications Chemistry*, 8:193, 2025. doi: 10.1038/s42004-025-01585-0. URL https://doi.org/10.1038/s42004-025-01585-0.

Hannes Stärk, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi S. Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46495–46513. PMLR, 2024. URL https://openreview.net/forum?id=syXFAVqx85.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. PepTune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59017–59065. PMLR, 2025. URL https://proceedings.mlr.press/v267/tang25n.html.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*. PMLR, 2022.

Randi Vita, Nina Blazeska, Daniel Marrama, Sebastian Duesing, Jason Bennett, Jason Greenbaum, Marcus De Almeida Mendes, Jarjapu Mahita, Daniel K. Wheeler, Jason R. Cantrell, James A. Overton, Darren A. Natale, Alessandro Sette, and Bjoern Peters. The immune epitope database (IEDB): 2024 update. *Nucleic Acids Research*, 53(D1):D436–D443, 2025. doi: 10.1093/nar/gkae1092. URL https://doi.org/10.1093/nar/gkae1092.

Shunzhou Wan, Xibei Zhang, Xiao Xue, and Peter V. Coveney. On the reliability of AI methods in drug discovery: Evaluation of Boltz-2 for structure and binding affinity prediction. *Journal of Chemical Theory and Computation*, 2026. doi: 10.1021/acs.jctc.6c01334. URL https://doi.org/10.1021/acs.jctc.6c01334.

Yinuo Zhang, Sophia Tang, Tong Chen, Elizabeth Mahood, Sophia Vincoff, and Pranam Chatterjee. PeptiVerse: A unified platform for therapeutic peptide property prediction. *Nature Communications*, 17:6819, 2026. doi: 10.1038/s41467-026-74167-w. URL https://doi.org/10.1038/s41467-026-74167-w.

Appendix

A1 Further Algorithmic Details

A1.1 NimblDrift Details

For neural embeddings, Algorithm A1 implements one sequential sweep of Equation (4). The objective is the empirical log-density ratio in Equation (3). The corresponding exact reverse KL between the smoothed feature densities would be

$$D_{\mathrm{KL}}(\widehat{q}_\tau \| \widehat{p}_\tau) = \int_{\mathbb{R}^d} \widehat{q}_\tau(z) \log \frac{\widehat{q}_\tau(z)}{\widehat{p}_\tau(z)} \, dz. \tag{9}$$

This integral averages the log-density ratio over Gaussian-perturbed candidate embeddings. The implemented score averages over the unperturbed embeddings and retains each candidate's contribution to its own density estimate. It therefore need not inherit nonnegativity or other divergence guarantees from exact KL. We use "reverse KL" in the plots as shorthand for this surrogate.

Following the cross-population distance scaling used in drifting (Deng et al., 2026), we set the bandwidth at the start of each sweep to

$$\tau_t = \tau_0 \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \|\phi(x_t^{(i)}) - \phi(y_j)\|_2, \qquad \tau_0 = 0.2. \tag{10}$$

The bandwidth equals the mean candidate-reference distance multiplied by the scale τ_0. It is fixed for all candidate comparisons within that sweep and recomputed at the next sweep. Thus, accepted updates cannot increase a fixed-sweep objective, but scores from consecutive sweeps use different bandwidths and need not decrease. The calculation assumes a positive mean cross-distance. We intend to release the complete implementation with the paper.

Algorithm A1 One NimblDrift sweep using the log-density-ratio surrogate and uniform reference weights.

```
def nimbldrift_step(X, Y, nbhd, phi, tau=0.2,
                    tau_normalization="cross-mean"):
    # X and Y are nonempty ordered candidate/reference lists.
    # phi(list) returns embeddings; nbhd(x) returns candidate edits.
    Z_Y = phi(Y)
    if tau_normalization == "cross-mean":
        tau = tau * cdist(phi(X), Z_Y).mean()
    # The same positive bandwidth is used throughout this sweep.
    def log_kde(Z_q, Z_s):
        d2 = ((Z_q[:, None] - Z_s[None, :]) ** 2).sum(-1)
        return logsumexp(-d2 / (2 * tau ** 2), dim=1) - log(len(Z_s))

    def density_ratio_score(Z_X):
        return (log_kde(Z_X, Z_X) - log_kde(Z_X, Z_Y)).mean()

    D, X_new = density_ratio_score(phi(X)), list(X)
    for i, x in enumerate(X):
        for c in nbhd(x):
            X_try = X_new[:i] + [c] + X_new[i + 1:]
            D_try = density_ratio_score(phi(X_try))
            if D_try < D:  # Retain the current candidate on ties.
                X_new[i], D = c, D_try
    return X_new
```

The Gaussian normalization constant cancels in the log-density ratio and is omitted in the pseudocode. For reference weights w_j, replace the reference log-density calculation by $\mathrm{logsumexp}_j[\log w_j - \|z - r_j\|_2^2/(2\tau^2)]$, using only positive-weight references. The candidate density remains uniformly weighted. Sequential acceptance uses the updated population for every comparison, and scanning the neighborhood retains the lowest-scoring candidate encountered for that coordinate.

For GreedyDrift, let $q' = q + M^{-1}(\delta_u - \delta_{x^{(i)}})$ replace candidate $x^{(i)}$ by u. The reward improvement and divergence minimization terms are balanced by a rank-normalized selection procedure.

$$\mathcal{J}_t(q') = \mathrm{rank}_{q_t, x^{(i)}}(\mathbb{E}_{x\sim q'} R(x)) - \lambda \cdot \mathrm{rank}_{q_t, x^{(i)}}(\mathcal{D}_t(q', p_{\mathrm{ref}})) \tag{11}$$

the $\mathrm{rank}_{\mathrm{q_t, x^{(i)}}}$ operator scores all possible edits, including the null edit, to candidate $x^{(i)}$ and returns the rank of the distribution q' amongst them. This normalization strategy simplifies selection of tradeoff parameter λ by operating invariant of the scale of R and $\mathcal{D}_t$.

For binary Morgan fingerprints $u, v \in \{0,1\}^d$, Tanimoto similarity is $k_{\mathrm{T}}(u,v) = u^\top v/(\|u\|_1 + \|v\|_1 - u^\top v)$, with $k_{\mathrm{T}}(0,0) = 1$. If $\mu = \phi_{\#}q$ and $\nu = \phi_{\#}p_{\mathrm{ref}}$, squared MMD is

$$\mathrm{MMD}^2_{k_{\mathrm{T}}}(\mu,\nu) = \mathbb{E}_{u,u'\sim\mu} k_{\mathrm{T}}(u,u') + \mathbb{E}_{v,v'\sim\nu} k_{\mathrm{T}}(v,v') - 2\mathbb{E}_{u\sim\mu, v\sim\nu} k_{\mathrm{T}}(u,v). \tag{12}$$

Each pair is sampled independently, and diagonal ($u = u'$, $v = v'$) terms are retained in the estimator. This expression compares within-population and cross-population similarities and can be evaluated as weighted sums for empirical populations. Minimizing MMD or its square gives the same candidate ordering for the exact empirical kernel-mean distance, but they are not the same reported number: Table 1 and the self-MMD benchmark both report MMD^2 (Eq. 12) directly, with no square root applied.

Runtime Summary. By default, NimblDrift requires computing embeddings for every neighbor of every member of its population at every step. Specifically, $\mathcal{O}(k \cdot N \cdot T)$ embeddings calls are required where k bounds the neighborhood size, N is the size of the drift population, and the algorithm returns for T steps. Though it is a compute intensive process, it still remains cheap relative to generative model training to generate a single batch of samples. All biologic drifts were conducted on a single H100 GPU. Table A1 reports the runtime of a single drift step, representative of each of the paper's main experiments.

Table A1: Wall-clock time for a single drift step (seconds), measured at each experiment's actual population size N. # steps and total time use each experiment's own reported # steps (Tables 3, 5, 7).

Experiment	Encoder	N	s/step	# steps	total time
Molecular Binder	SMI-TED$_{289M}$	100	9.8	40	392 s (6.5 min)
Molecular Binder	Molformer	100	7.9	40	316 s (5.3 min)
Molecular Binder	Morgan FP	100	6.2	40	248 s (4.1 min)
Therapeutic Peptides (12-mer)	ESM-2 35M	407	6.4	15	96 s (1.6 min)
Therapeutic Peptides (12-mer)	PeptideCLM-2	407	72.0	15	1080 s (18.0 min)
MHCI Eluted Ligand (9-mer)	ESM-2 35M	200	2.1	20	41 s (0.7 min)
MHCII Eluted Ligand (15-mer)	ESM-2 35M	200	3.9	20	78 s (1.3 min)

A1.2 Crossover Neighborhood Implementation

For molecular crossovers, we used either a fixed pool of 10,000 BindingDB molecules (Liu et al., 2007; 2025) or the current drift population as the partner buffer. Algorithm A2 samples $K = 10$ partners without replacement, attempts both parental orders, and applies the Graph GA mutation subroutine with per-child probability $\rho = 0.067$ (Jensen, 2019). Failed crossover proposals are discarded, and valid products are canonicalized and deduplicated. Thus, K counts sampled partners, and the number of retained offspring can differ from K.

Algorithm A2 Genetic neighborhood $\mathcal{N}_{B\text{-crossover}}(x)$ for one molecule.

```
def genetic_nbhd(x, pool, K=10, rho=0.067):
    # x is a parent molecule; pool contains partner molecules.
    # K is the number of partners; rho is the mutation probability.

    mates = sample(pool, K, replace=False)    # draw K candidate mates
    children, seen = [], {x}
    for w in mates:
        for a, b in [(x, w), (w, x)]:         # both breeding orders
            c = crossover(a, b)               # cut + reassemble; None on
    failure
            if c is None: continue
            c = canonical(mutate(c, rho))     # mutate w.p. rho, then
    canonicalize
            if c not in seen:
                seen.add(c); children.append(c)

    return [x] + children                     # include self (candidate
    may stay put)
```

A2 Empirical Analysis

A2.1 Kleinberg's Small-World Network

The synthetic comparison in Section 4 used a lattice with long-range edges sampled by Kleinberg's construction (Kleinberg, 2000). Figure A1 depicts the same graph without trajectory overlays.

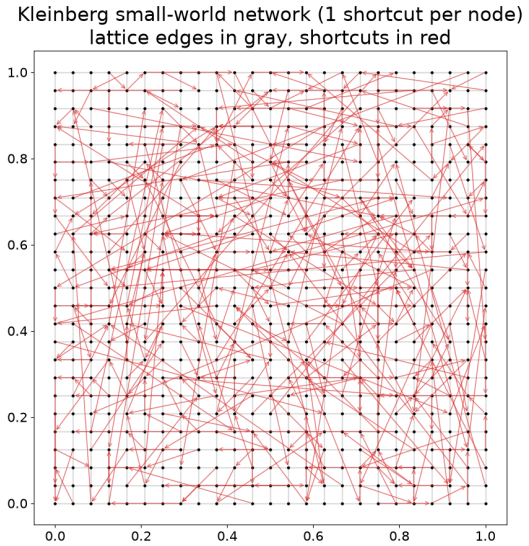

Figure A1: **Underlying Graph of Figure 1**

A2.2 Projected Continuous Drift

To test the relationship between discrepancy reduction and agreement with continuous trajectories in peptide space, we repeated the A*01:01 reference experiment from Section 5.4 using Equations (4) and (7). The projected update plateaued at a higher log-density-ratio score. More of its terminal candidates nevertheless shared their nearest reference neighbor with the corresponding continuous trajectories (Figure A2). This pattern was consistent with the synthetic-network comparison, where direct discrepancy minimization reached lower scores with less agreement in individual endpoints.

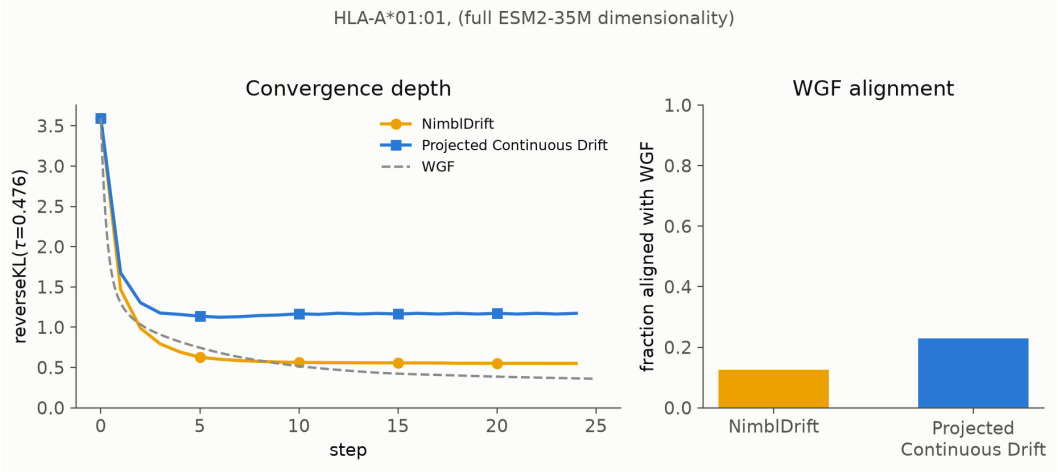

Figure A2: **NimblDrift and projected continuous drift in peptide space.** *Left.* Direct population updates attained a lower log-density-ratio score. *Right.* A greater fraction of projected-update endpoints shared their nearest reference neighbor with the corresponding continuous trajectories. The reverse-KL plot label denotes the surrogate in Equation (3).

A2.3 Crossover Neighborhood Convergence Experiment

We initialized 500 molecules from BindingDB and applied drift toward a reference of 184 antibiotics from Scalia et al. (2025), comparing Morgan-fingerprint MMD^2 under the three neighborhoods in Table 1. The reference self-MMD^2 value of approximately 0.0065 is a finite-sample comparison benchmark, not an optimization lower bound. Crossover augmentation also accelerated convergence in the subsequent Molformer and SMI-TED_{289M} binder-generation experiments.

A3 Further Experimental Details

This appendix records additional choices for the biological experiments and examines the effects of population size, reference composition, and peptide residue order.

A3.1 Small-Molecule Binder Generation

Experimental Setup. We compared the molecular representations and neighborhoods summarized in Table A2.

Table A2: Experimental setup for small-molecule binder generation.

Component	Choice	Justification
Initial Population	100 random BindingDB ligands	Initial compounds were sampled without selecting for binding to the evaluation target.
Reference Population	Literature binders, with an expanded reference adding 200 ligands from target-specific BindingDB assays	We compared small literature-derived references with larger assay-derived references.
Embedding / Discrepancy	{SMI-TED_{289M}, Molformer, Morgan Fingerprints} with the discrepancies in Section 3.1	Molformer has been evaluated for molecular representation geometry (Dabrowski-Tumanski et al., 2026). Comparing it with SMI-TED_{289M} tests how encoder choice affects reference-based generation.

continued on next page

Table A2 (continued)

Component	Choice	Justification
Neighborhood	$\{\mathcal{N}_{\text{graph-edit}}, \mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{BDB-crossover}}, \mathcal{N}_{\text{graph-edit}} \cup \mathcal{N}_{\text{GA-crossover}}\}$, $K = 10$ crossovers	We tested whether the convergence improvements in Section 4.3 corresponded to improved binding scores.
Reward Signal	None	
Evaluation Metrics	Binding likelihood (p_{bind}) as scored by Boltz-2 (Passaro et al., 2025)	Boltz-2 offers a structure-based evaluation score with published benchmarking (Wan et al., 2026; Grieswelle et al., 2026).
Baseline	FlashBind (Jiang et al., 2026) greedy optimization, random walk with crossover proposals	FlashBind was also used for comparison in the source benchmark (Chen et al., 2026).
Timesteps	40	

Further Details. The "known binders" for each protein target are the co-crystallized ligand plus a small set of literature binders from Chen et al. (2026): for 2VT4 (beta-1 adrenergic receptor), cyanopindolol (co-crystallized), carmoterol, isoprenaline, dobutamine, salbutamol, carvedilol, carazolol, bisoprolol, and ICI-118551; for 6CM4 (dopamine D2 receptor), risperidone (co-crystallized) and haloperidol; and for 5SDV (PDE10A), its co-crystallized ligand (PDB code IAI) plus papaverine, MP-10, PQ-10, dipyridamole, SCH51866, and IBMX.

For a fixed protein target and reference ligands $\{y_j\}_{j=1}^{N}$, we normalize their Boltz-2 scores to obtain the weighted empirical reference,

$$w_j = \frac{p_{\text{bind}}(y_j)}{\sum_{\ell=1}^{N} p_{\text{bind}}(y_\ell)}, \qquad p_{\text{ref}} = \sum_{j=1}^{N} w_j \delta_{y_j}. \tag{13}$$

The weights sum to one and increase the contribution of higher-scoring references, assuming a positive denominator. We use these weights in the reference density of Equation (2). The unweighted condition uses $w_j = 1/N$.

To analyze the properties of the *tradeoff* between generalization and binding scores in the terminal drift populations, we plot the pareto frontiers for every encoder and neighbor configuration in Figure A3. We use the median Morgan-Tanimoto distance to the nearest member of the reference population as a metric of generalization, as it is estasblished in the literature (Dabrowski-Tumanski et al., 2026). We find that there is a clear trade off between the mean p_{bind} score and generlization. In 2VT4 and 5SDV all drifts land safely away from the self-reference nearest neighbor baseline. In 6CM4 the SMI-TED$_{289M}$ drift lands within, but this is primarily due to the reference self-distance being larger indicating that the reference population itself is more diverse.

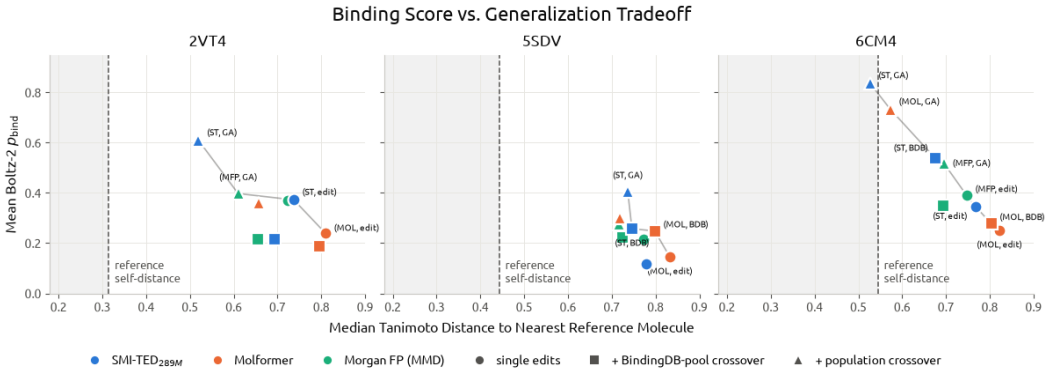

Figure A3: **Binding Score vs. Generalization for Small Molecule Drifts.** We plot the pareto frontier for drifts towards the full known binder + BindingDB populations using the {SMI-TED$_{289M}$, Molformer, Morgan FP} feature space along with the $\{\mathcal{N}_{\text{graph-edit}}, \mathcal{N}_{\text{BDB-crossover}}, \mathcal{N}_{\text{GA-crossover}}\}$ neighborhood.

In a one-off experiment we found that scaling the population size of the Morgan FP drift using $\mathcal{N}_{\text{GA-crossover}}$ and the BindingDB population led to far improved performance in binding scores. Specifically, we ran the drift with $N = 1,800$ population members and saw that 2VT4 mean p_{bind} increased from 0.355 to 0.685, 5SDV mean p_{bind} increased from 0.291 to 0.599, and 6CM4 mean p_{bind} increased from 0.439 to 0.664. This came at a large cost to generalization. The terminal drift iterates included $81, 158, 151$ exact copies of reference molecules of the terminal population. We suspect that this is likely due to the close structural relationship between Morgan FP and the graph-structures themselves. The impacting of increasing population sizes on representation encoder drifts remains to be studied.

A3.2 Therapeutic Regularization in Peptide Design

Experimental Setup. We compared reference-based regularization with greedy search and MOG-DFM using the setup in Table A3.

Table A3: Experimental setup for therapeutic regularization in peptide binding optimization.

Component	Choice	Justification
Initial Population	400 uniformly random 12-mers	
Reference Population	400 naturally occurring 12-mers from the PepDFM (Chen et al., 2025) training corpus	Fixed-length sequences limit length-related differences in mean-pooled representations.
Embedding / Discrepancy	{ESM-2 35M, PeptideCLM-2} mean-pooled embedding (Lin et al., 2023; Feller et al., 2026)	We compared a protein encoder with a peptide-focused encoder.
Neighborhood	$\mathcal{N}_{\text{single-substitution}}$	
Reward Signal	PeptiVerse binding-affinity predictor (Zhang et al., 2026)	Affinity was the only directly optimized property score.
Evaluation Metrics	PeptiVerse non-fouling, solubility, and hemolysis predictors	
Baseline	Unregularized greedy optimization, MOG-DFM (Chen et al., 2025)	The comparisons include unregularized search and guidance of a trained generative prior.
Timesteps	15	

A3.3 Eluted Ligand Generation for the pMHC Pathway

Experimental Setup. We evaluated class I and class II presentation using the allele-specific references and human-proteome controls in Table A4.

Table A4: Experimental setup for eluted ligand generation in the pMHC pathway.

Component	Choice	Justification
Initial Population	200 uniformly random 9-mers (Class I) or 15-mers (Class II)	We fixed peptide length within each MHC class to control length-dependent effects.
Reference Population	200 allele-specific IEDB EL peptides. Separate controls used length-matched human-proteome peptides. Reference records were selected by entry date after the stated predictor-training cutoff.	Allele-associated EL measurements define the reference populations. Date filtering alone does not exclude sequence overlap with earlier predictor training data.
Embedding	ESM-2 35M mean-pooled embedding (Lin et al., 2023)	

continued on next page

Table A4 (continued)

Component	Choice	Justification
Neighborhood	$\mathcal{N}_{\text{single-substitution}}$	
Reward Signal	None	
Evaluation Metrics	Change in median EL %Rank, NetMHCpan-4.1 (class I) / NetMHCIIpan-4.0 (class II) (Reynisson et al., 2020)	Lower percentile ranks indicate stronger predicted presentation.
Baseline	None	
Timesteps	20	

Further Details. The reference populations for all drifts were pulled from the IEDB API with a date cutoff after 2021. This year was selected because the NetMHCpan-4.1 (Class I) and NetMHCIIpan-4.0 (Class II) both have a 2020 training data cutoff (Reynisson et al., 2020).

We repeated each allele-specific drift across six seeds, resampling both initial and reference populations. Let $r_b(x)$ be the EL percentile rank assigned to peptide x by the predictor for evaluation allele b, and let $\mathcal{Q}_{T,a}$ and $\mathcal{Q}_{0,a}$ denote the terminal and initial populations for reference allele a. For each run, the specificity-matrix entry is

$$\Delta_{a,b} = \text{median}_{x \in \mathcal{Q}_{T,a}} r_b(x) - \text{median}_{x \in \mathcal{Q}_{0,a}} r_b(x). \tag{14}$$

This difference measures the change in median predicted presentation rank, with negative values indicating improvement. Diagonal entries use the same reference and evaluation allele. The human-proteome control is evaluated identically.

To measure the level of generalization in the terminal populations, we look the median Hamming distance between terminal members and their nearest-neighbor in the reference. We find that no drift decreases the median nearest-neighbor hamming distance more than 0.167 in every configuration. Specifically, all Class I drifts start and remain at median nearest-neighbor hamming distance of 6.0 except "HLA-B*07:02" which moves from 6.167 to 6.0. Similarly all Class II drifts start and end at 11.0 except "HLA-DRB1*01:01" which moves from 11.167. This result indicates that ESM-2 representation is not simply encoding sequence level statistics in it's geometric prior.

Further Experiment. Do the terminal populations retain position-dependent sequence information beyond amino-acid composition? We compared the terminal and reference populations with versions in which each peptide's residue order was randomly shuffled. This preserves each peptide's amino-acid composition while disrupting residue ordering.

Scrambling the terminal peptides worsened diagonal EL percentile ranks, consistent with a contribution from sequence order to predicted presentation (Figures A4 and A5). Scrambled references had the weakest presentation scores in this comparison, indicating a larger effect of shuffling on reference peptides. Both scrambled and unmodified generated populations retained allele-associated patterns. Thus, the scrambling analysis supports an ordering-dependent score component, while composition can also contribute to allele discrimination. Sequence novelty relative to the reference requires a separate comparison.

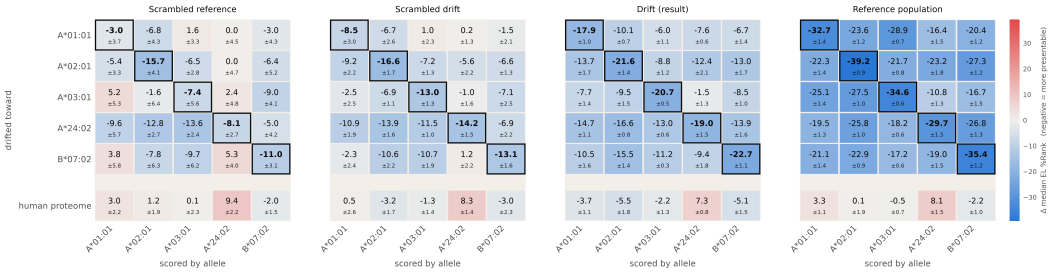

Figure A4: **MHC class I sequence-scrambling comparison.** Panels compare scrambled reference peptides, scrambled generated peptides, unmodified generated peptides, and reference peptides. Cells report changes in median EL percentile rank, with more negative values indicating stronger predicted presentation. Residue shuffling preserves each peptide's amino-acid composition.

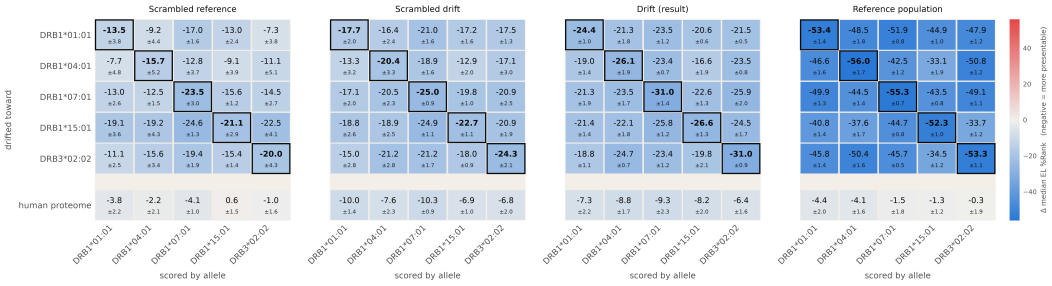

Figure A5: **MHC class II sequence-scrambling comparison.** The four conditions and percentile-rank convention match Figure A4. Comparisons between shuffled and unmodified sequences assess the contribution of residue order to the predicted presentation score.