Multi-Objective-Guided Discrete Flow Matching for Controllable Biological Sequence Design

Tong Chen [1] **Yinuo Zhang** [2] **Sophia Tang** [1] **Pranam Chatterjee** [1 3]

Abstract

We present **Multi-Objective-Guided Discrete Flow Matching (MOG-DFM)**, a general framework to steer any pretrained discrete-time flow matching generator toward Pareto-efficient trade-offs across multiple scalar objectives. At each sampling step, MOG-DFM computes a hybrid rank-directional score for candidate transitions and applies an adaptive hypercone filter to enforce consistent multi-objective progression. We also trained two unconditional discrete flow matching models, **PepDFM** for diverse peptide generation and **EnhancerDFM** for functional enhancer DNA generation, as base generation models for MOG-DFM. We demonstrate MOG-DFM's effectiveness in generating peptide binders optimized across five properties (hemolysis, non-fouling, solubility, half-life, and binding affinity), and in designing DNA sequences with specific enhancer classes and DNA shapes. In total, MOG-DFM proves to be a powerful tool for multi-property-guided biomolecule sequence design.

1. Introduction

Designing biological sequences that simultaneously satisfy multiple functional and biophysical criteria is a foundational challenge in modern bioengineering (Naseri & Koffas, 2020; Tominaga et al., 2024; Mohr et al., 2016; Schmidt et al., 2025; Artemyev et al., 2024). Most existing biomolecule-design methods focus on optimizing a single objective in isolation (Zhou et al., 2019; Nehdi et al., 2020). For example, efforts have been made to reduce protein toxicity (Kreiser et al., 2020; Sharma et al., 2022) and neural networks are used to improve protein thermo-stability (Komp et al., 2025). While these single-objective approaches yield high performance on their target metrics, they often produce sequences with undesirable trade-offs—high-affinity peptides may be insoluble or toxic, and stabilized proteins may lose functional specificity (Bigi et al., 2023; Rinauro et al., 2024). Consequently, a framework for multi-objective guided generation that can balance conflicting requirements is critical to meet the demands of biomolecular engineering.

Classical multi-objective optimization (MOO) techniques, such as evolutionary algorithms and Bayesian optimization, have been successfully applied to black-box tuning of molecular libraries (Zitzler & Thiele, 1998; Deb, 2011; Ueno et al., 2016; Frisby & Langmead, 2021). More recently, controllable generative models have been developed to integrate MOO directly into the sampling process (Li et al., 2018; Sousa et al., 2021; Yao et al., 2024). ParetoFlow (Yuan et al., 2024), for instance, leverages continuous-space flow matching to produce Pareto-optimal samples, but operates only in continuous domains. Applying such techniques to discrete sequences typically requires embedding into a continuous manifold, which can distort distributions and complicate property-based guidance (Beliakov & Lim, 2007; Michael et al., 2024).

Discrete flow matching has recently emerged as a powerful paradigm for directly modeling and sampling from complex discrete spaces (Gat et al., 2024; Dunn & Koes, 2024). Two primary variants exist: (i) continuous-time simplex methods, which diffuse discrete data through a continuous embedding over the probability simplex (Stark et al., 2024; Davis et al., 2024; Tang et al., 2025), and (ii) jump-process models that learn time-dependent transition rates for token-level stochastic updates (Gat et al., 2024). The latter is particularly well suited for controllable generation, as it naturally supports reweighting of token transitions based on reward functions.

Recent work has applied these models to single-objective tasks: Nisonoff et al. (2025) introduced rate-based classifier guidance for pretrained samplers, while Tang et al. (2025) proposed Gumbel-Softmax Flow Matching with straight-through guidance for controllable discrete generation. Yet, to our knowledge, no prior work has extended discrete flow matching to support Pareto-guided generation across multi-

[1]Department of Computer and Information Science, University of Pennsylvania [2]Center of Computational Biology, Duke-NUS Medical School [3]Department of Bioengineering, University of Pennsylvania. Correspondence to: Pranam Chatterjee <pranam@seas.upenn.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

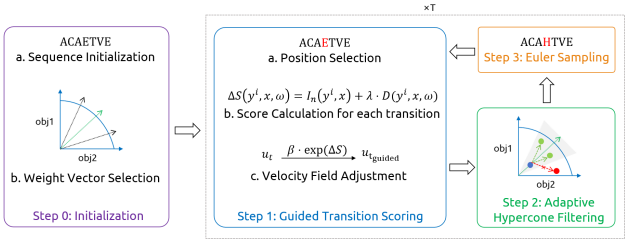

Figure 1. Visualization for MOG-DFM algorithm.

ple objectives.

As such, our key contributions are as follows:

1. **MOG-DFM: Multi-Objective-Guided Discrete Flow Matching**, a general framework that steers pre-trained discrete flow matching models toward Pareto-efficient solutions via multi-objective guidance and adaptive hypercone filtering.
2. **Rank-Directional Scoring and Hypercone Filtering** combine rank-normalized local improvement and directional alignment with a user-specified trade-off vector to reweight token-level transition velocities, followed by a dynamic angular filtering mechanism that enforces directional consistency.
3. **Unconditional Base Models for Biomolecule Generation**; we train two high-quality discrete flow matching models—**PepDFM** for diverse peptide generation and **EnhancerDFM** for functional enhancer DNA generation—demonstrating low loss and biological plausibility.
4. **Multi-Property Sequence Design**; we apply MOG-DFM to two challenging biological generation tasks: (i) therapeutic peptide binder generation with five competing objectives (affinity, solubility, hemolysis, half-life, non-fouling), and (ii) enhancer DNA sequence generation guided by enhancer class and DNA shape.

2. MOG-DFM

MOG-DFM operates under the same setting as discrete flow matching described in Appendix B. Suppose we have a pre-trained discrete flow matching model that defines a CTMC with a factorized velocity field $u_t^i(y^i, x)$, which transports probability mass from an initial distribution p_0 to the unknown target distribution via mixture path parametrization. In addition, we assume access to N pre-trained scalar score functions $s_n : \mathcal{S} \to \mathbb{R}$, where $n = 1, \ldots, N$, that assign objective scores to any sequence. Our aim is to generate novel sequences $x_1 \in \mathcal{S}$ whose objective vectors $\big(s_1(x_1),\, s_2(x_1),\, \ldots, s_N(x_1)\big)$ lie near the Pareto front (not guaranteed to be Pareto optimal)

$$\mathrm{PF} \;=\; \big\{x \in \mathcal{S} \;\big|\; \nexists\, x' \in \mathcal{S} :\; s_n(x') \geq s_n(x)\ \forall n,\ \exists\, m : s_m(x') > s_m(x)\big\}.$$

To achieve this, we will guide the CTMC sampling dynamics of the discrete flow matching model using multi-objective transition scores, steering the generative process toward Pareto-efficient regions of the state space (Figure 1, Pseudo code 1, Proof in Section L).

MOG-DFM begins by initializing the generative process at time $t = 0$ by sampling an initial sequence x_0 uniformly from the discrete state space $\mathcal{S} = [K]^d$. A weight vector ω is then generated to specify a direction to optimize in the state space, as detailed in Appendix A. The following three steps will then be performed in each iteration. We set the number of total iterations to be T.

2.1. Step 1: Guided Transition Scoring

We first randomly select one position i on the sequence so that we will update the token on this position during the current iteration. At each intermediate state x_t and selected position i, each possible candidate transition $y^i \neq x^i$ is scored by combining local improvement measures with global directional alignment. The normalized rank score captures how much each individual objective improves relative to other possible token replacements, thereby encouraging exploration of promising local moves; formally, for each objective n we compute

$$I_n(y^i, x) = \frac{\mathrm{rank}\big(s_n(x_{\text{new}}) - s_n(x)\big)}{|T|}, \tag{1}$$

where x_{new} denotes the sequence obtained by replacing the ith token of x with y^i. The rank$(\cdot)$ function maps the raw score change into a uniform scale in $[0, 1]$. In contrast, the directional term

$$D(y^i, x, \omega) = \Delta\mathbf{s}(y^i, x) \cdot \omega \tag{2}$$

measures the alignment of the multi-objective improvement vector $\Delta\mathbf{s}$ with the chosen weight vector ω, ensuring that transitions not only improve individual objectives but collectively move toward the desired trade-off direction. By z-score normalizing both components and combining them as

$$\Delta S(y^i, x, \omega) = \mathrm{Norm}\Big[\tfrac{1}{N}\textstyle\sum_{n=1}^{N} i_n I_n(y^i, x)\Big] + \lambda\,\mathrm{Norm}\big[D(y^i, x, \omega)\big], \tag{3}$$

we balance rank-based exploration against direction-guided exploitation with $\lambda > 0$. An importance vector $\mathbf{I} = [i_1, \ldots, i_N]$ is used to normalize the improvement values for each objective. Finally, we re-weight the original factorized velocity field from the pre-trained discrete flow matching model:

$$u^i_{\text{guided},t}(y^i, x \mid \omega) = \begin{cases} \beta\, u_t^i(y^i, x)\, \exp\big(\Delta S(y^i, x, \omega)\big), & y^i \neq x^i \\ -\sum_{y^i \neq x^i} u^i_{\text{guided},t}(y^i, x \mid \omega), & y^i = x^i \end{cases} \tag{4}$$

where β is the strength hyperparameter. Therefore, the guided velocities satisfy the non-negativity and zero-sum rate conditions by construction, preserving valid CTMC dynamics while favoring high-utility transitions.

2.2. Step 2: Adaptive Hypercone Filtering

To ensure each candidate token replacement drives the sequence towards the chosen trade-off direction, we restrict candidate transitions to lie within a cone around the weight vector ω. This "hypercone" mechanism allows the sampler to navigate non-convex or discontinuous regions of the Pareto front by enforcing local directional consistency. Specifically, for a given position i and candidate token y^i, we compute the angle

$$\alpha^i = \arccos\left(\frac{\Delta\mathbf{s}(y^i, x) \cdot \omega}{\|\Delta\mathbf{s}(y^i, x)\| \, \|\omega\|}\right), \tag{5}$$

where $\Delta\mathbf{s}(y^i, x)$ is the multi-objective improvement vector from replacing x^i with y^i. We accept only those y^i for which $\alpha^i \leq \Phi$, where Φ denotes the current hypercone angle. Denoting $Y^i \subseteq T \setminus \{x^i\}$ as the set of accepted tokens, we select the best transition as

$$y^i_{\text{best}} = \arg\max_{y^i \in Y^i} \Delta S(y^i, x, \omega) \quad \text{if } Y^i \neq \emptyset. \tag{6}$$

There are two degenerate cases that can lead to empty Y^i: If every $\alpha^i \geq \pi$, indicating that all possible transitions decrease performance, we will perform a self-transition and retain the current state; if there exist some $\alpha^i < \pi$ but none lie within the cone (i.e. Φ is temporarily too small), we still advance by choosing the best-aligned candidate

$$y^i_{\text{best}} = \arg\max_{\{y' : \alpha^i < \pi\}} \Delta S(y^i, x, \omega), \tag{7}$$

allowing progress while the hypercone angle self-adjusts.

As a pre-defined hypercone angle may be too big or too small during the dynamic optimization process, we need to adaptively tune the angle that best balances exploration and exploitation. Specifically, we compute the rejection rate

$$r_t = \frac{\#\{y^i : \alpha^i > \Phi\}}{\text{total \# of candidate transitions}} \tag{8}$$

and its exponential moving average (EMA)

$$\bar{r}_t = \alpha_r \, \bar{r}_{t-h} + \big(1 - \alpha_r\big) \, r_t, \tag{9}$$

where $\alpha_r \in [0, 1)$ is a smoothing coefficient and $\bar{r}_0 = \tau$ is the target rejection rate. We then update the hypercone angle via

$$\Phi_{t+h} = \text{clip}\Big(\Phi_t \, \exp\big(\eta \, (\bar{r}_t - \tau)\big), \, \Phi_{\min}, \, \Phi_{\max}\Big), \tag{10}$$

with learning rate $\eta > 0$ and bounds $\Phi_{\min}, \Phi_{\max}$ to prevent the hypercone from collapsing or over-expanding. Intuitively, if too many candidates are being rejected ($\bar{r}_t > \tau$), the hypercone widens to admit more directions; if too few are rejected ($\bar{r}_t < \tau$), it narrows to focus on the most aligned transitions.

2.3. Step 3: Euler Sampling

Once the guided transition rates $u^i_{\text{guided},t}(y^i, x \mid \omega)$ have been computed and the best candidate transition has been selected after hypercone filtering (if not self-transitioning), we evolve the CTMC via Euler sampling. We denote the total outgoing rate from x at time t on coordinate i by

$$R^i_t(x) = -\, u^i_{\text{guided},t}(x^i, x \mid \omega) = \textstyle\sum_{y^i \neq x^i} u^i_{\text{guided},t}(y^i, x \mid \omega). \tag{11}$$

The one-step transition kernel for coordinate i is given by the exact Euler–Maruyama analogue for CTMCs:

$$\mathbb{P}\big(X^i_{t+h} = y^i \mid X_t = x\big) = \begin{cases} \exp\big(h \, u^i_{\text{guided},t}(x^i, x \mid \omega)\big) = \exp\big(-h \, R^i_t(x)\big), & y^i = x^i, \\ \dfrac{u^i_{\text{guided},t}(y^i, x \mid \omega)}{R^i_t(x)} \big(1 - \exp(-h \, R^i_t(x))\big), & y^i \neq x^i. \end{cases} \tag{12}$$

Here, $h = 1/T$ is the step size in the time interval, X_t and X_{t+h} denotes the current state and the next state respectively. In practice, one draws a uniform random number $r \in [0, 1]$: if $r \leq 1 - \exp(-h \, R^i_t(x))$, x^i will transition to the best selected candidate; otherwise we retain x^i.

After performing from step 1 to step 3 for T iterations, we end with the final sample x_1 whose score vectors have been steered close to the Pareto Front, with all objectives optimized.

3. Experiments

To the best of our knowledge, there are no public datasets that serve to benchmark multi-objective optimization algorithms for biological sequences. Therefore, we develop two benchmarks to evaluate MOG-DFM: multi-objective guided peptide binder sequence generation and multi-objective guided enhancer DNA sequence generation. The performance of MOG-DFM's base models, PepDFM and EnhancerDFM, is detailed in Appendix C.

3.1. MOG-DFM generates peptide binders under five property guidance

We benchmark MOG-DFM on a peptide binder generation task guided by five different properties that are critical for therapeutic discovery: hemolysis, non-fouling, solubility, half-life, and binding affinity. To evaluate MOG-DFM in a controlled setting, we designed 100 peptide binders per target for ten diverse proteins—structured targets with known binders (1B8Q, 1E6I, 3IDJ, 5AZ8, 7JVS), structured targets without known binders (AMHR2, OX1R, DUSP12), and in-

trinsically disordered targets (EWS::FLI1, MYC) (Table 3). Across all targets and across multiple binder lengths, the generated peptides achieve low hemolysis rates (0.06–0.09), high non-fouling (>0.78) and solubility (>0.74), extended half-life (28–47 h), and strong affinity scores (6.4–7.6), demonstrating both balanced optimization and robustness to sequence length.

For the target proteins with pre-existing binders, we compared the property values between their known binders with MOG-DFM-designed ones (Figure 2, 5). The designed binders significantly outperform the pre-existing binders across all properties without compromising the binding potential, which is further confirmed by the ipTM scores computed by AlphaFold3 (Abramson et al., 2024) and docking scores calculated by AutoDock VINA (Trott & Olson, 2010). Although the MOG-DFM-designed binders bind to similar target positions as the pre-existing ones, they differ significantly in sequence and structure, demonstrating MOG-DFM's capacity to explore the vast sequence space for optimal designs. For target proteins without known binders, complex structures were visualized using one of the MOG-DFM-designed binders (Figure 6). The corresponding property scores, as well as ipTM and docking scores, are also displayed. Some of the designed binders demonstrated extended half-life, while others excelled in non-fouling and solubility, underscoring the comprehensive exploration of the sequence space by MOG-DFM.

At each iteration, we recorded the mean and standard deviation of the five property scores across all the 100 peptides of length 12 designed for EWS::FLI1 to evaluate the effectiveness of the guided generation strategy (Figure 4A). All five properties exhibited an improving trend over iterations, with the average score of the solubility and non-fouling properties showing a significant increase from a score around 0.3 to 0.8. A large deviation of the final half-life values is caused by the susceptibility of the half-life value to guidance, with MOG-DFM balancing the trade-offs between half-life and other values. The improvements of hemolysis, non-fouling, and solubility gradually converge, demonstrating MOG-DFM's efficiency in steering the generation process to the Pareto Front within only 100 iterations.

We visualized the distribution change steered by MOG-DFM by plotting the property score distribution of 100 peptides of length 12 designed for EWS::FLI1 and 100 peptides of the same length sampled unconditionally from PepDFM (Figure 4B). MOG-DFM effectively shifted and concentrated the peptide distribution so that the peptides possess improved properties for all the objectives, demonstrating MOG-DFM's ability to steer the generation so that all properties are optimized simultaneously.

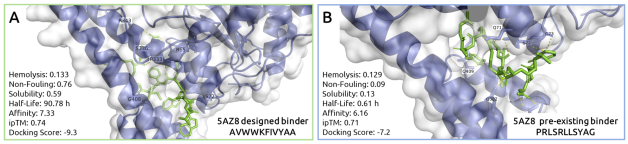

Figure 2. Complex structures of PDB 5AZ8 with a MOG-DFM-designed binder and its pre-existing binder. Five property scores are shown for each binder, along with the ipTM score from AlphaFold3 and docking score from AutoDock VINA. Interacting residues on the target are visualized.

3.2. MOG-DFM generates enhancer DNA of specific class with specified DNA shapes

To demonstrate the universal capability of MOG-DFM in performing multi-objective guided generation for biological sequences, we applied MOG-DFM to design enhancer DNA sequences guided by enhancer class and DNA shape. EnhancerDFM was used as the unconditional enhancer DNA sequence generator, while Deep DNAshape was employed to predict DNA shape (Li et al., 2024), and the enhancer class predictor from which it was sourced (Stark et al., 2024). Two distinct tasks with different enhancer class and DNA shape guidance were carried out, and ablation results are presented in Table 9. Given the time constraints, we designed five enhancer sequences of length 100 for each setting.

In the first task, we conditioned the generation to target enhancer class 1 (associated with the transcription factor binding motif ATF) and a high HelT (helix twist) value, with the maximum HelT value set to 36. With both guidance criteria in place, MOG-DFM effectively steered the sequence generation towards enhancer class 1 while simultaneously ensuring that the HelT value approached its maximum (Table 9). When one or both guidance criteria were removed, the corresponding properties showed significant degradation, with the probability of achieving the desired enhancer class dropping near zero (Table 9). A similar outcome was observed in the second task, which targeted enhancer class 16 and a higher Rise shape value, with the maximum Rise value set to 3.7. Since the canonical range for the Rise shape value spans from 3.3 to 3.4, MOG-DFM ensured both a high probability for the target enhancer class and an optimal DNA shape value, outperforming other ablation settings (Table 9). These results validate MOG-DFM's efficacy in multi-objective guided generation for DNA sequences.

4. Conclusion

In this work, we have presented **Multi-Objective-Guided Discrete Flow Matching (MOG-DFM)**, a scalable framework for generating biomolecular sequences that simultaneously optimize multiple, often conflicting properties. By guiding discrete flow matching models with multi-objective optimization, MOG-DFM can design peptide and DNA sequences with improved therapeutic and structural features.

While excelling in the biological domain, future work will extend MOG-DFM to other applications, including text and image generation. From a theoretical perspective, improving Pareto convergence guarantees and incorporating uncertainty-aware or feedback-driven guidance remain key directions to explore. Ultimately, MOG-DFM offers a foundation for generating the next generation of therapeutics—molecules that are not only effective but optimized for the multifaceted properties critical to clinical success.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abdin, O., Nim, S., Wen, H., and Kim, P. M. Pepnn: a deep attention model for the identification of peptide binding sites. *Communications biology*, 5(1):503, 2022.

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630 (8016):493–500, 2024.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019.

Artemyev, V., Gubaeva, A., Paremskaia, A. I., Dzhioeva, A. A., Deviatkin, A., Feoktistova, S. G., Mityaeva, O., and Volchkov, P. Y. Synthetic promoters in gene therapy: Design approaches, features and applications. *Cells*, 13 (23):1963, 2024.

Atak, Z. K., Taskiran, I. I., Demeulemeester, J., Flerin, C., Mauduit, D., Minnoye, L., Hulselmans, G., Christiaens, V., Ghanem, G.-E., Wouters, J., et al. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. *Genome research*, 31(6):1082–1096, 2021.

Beliakov, G. and Lim, K. F. Challenges of continuous global optimization in molecular structure prediction. *European journal of operational research*, 181(3):1198–1213, 2007.

Beume, N., Naujoks, B., and Emmerich, M. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European journal of operational research*, 181(3): 1653–1669, 2007.

Bigi, A., Lombardo, E., Cascella, R., and Cecchi, C. The toxicity of protein aggregates: new insights into the mechanisms, 2023.

Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., and Greenleaf, W. J. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, 10(12):1213–1218, 2013.

Bushuiev, A., Bushuiev, R., Kouba, P., Filkin, A., Gabrielova, M., Gabriel, M., Sedlar, J., Pluskal, T., Damborsky, J., Mazurenko, S., et al. Learning to design protein-protein interactions with enhanced generalization. *arXiv preprint arXiv:2310.18515*, 2023.

Coello, C. C. and Lechuga, M. S. Mopso: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pp. 1051–1056. IEEE, 2002.

Davis, O., Kessler, S., Petrache, M., Ceylan, I., Bronstein, M., and Bose, J. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

Deb, K. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34. Springer, 2011.

Deb, K. and Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

Dunn, I. and Koes, D. R. Exploring discrete flow matching for 3d de novo molecule generation. *ArXiv*, pp. arXiv–2411, 2024.

D'Aloisio, V., Dognini, P., Hutcheon, G. A., and Coxon, C. R. Peptherdia: database and structural composition analysis of approved peptide therapeutics and diagnostics. *Drug Discovery Today*, 26(6):1409–1419, 2021.

Frisby, T. S. and Langmead, C. J. Bayesian optimization with evolutionary and structure-based regularization for directed protein evolution. *Algorithms for Molecular Biology*, 16(1):13, 2021.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385, 2024.

Guntuboina, C., Das, A., Mollaei, P., Kim, S., and Barati Farimani, A. Peptidebert: A language model based on transformers for peptide property prediction. *The Journal of Physical Chemistry Letters*, 14(46):10427–10434, 2023.

Jain, S., Gupta, S., Patiyal, S., and Raghava, G. P. Thpdb2: compilation of fda approved therapeutic peptides and proteins. *Drug Discovery Today*, pp. 104047, 2024.

Komp, E., Phillips, C., Lee, L. M., Fallin, S. M., Alanzi, H. N., Zorman, M., McCully, M. E., and Beck, D. A. Neural network conditioned to produce thermophilic protein sequences can increase thermal stability. *Scientific Reports*, 15(1):14124, 2025.

Kreiser, R. P., Wright, A. K., Block, N. R., Hollows, J. E., Nguyen, L. T., LeForte, K., Mannini, B., Vendruscolo, M., and Limbocker, R. Therapeutic strategies to reduce the toxicity of misfolded protein oligomers. *International journal of molecular sciences*, 21(22):8651, 2020.

Li, J., Chiu, T.-P., and Rohs, R. Predicting dna structure using a deep learning method. *Nature communications*, 15(1):1243, 2024.

Li, Y., Zhang, L., and Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10:1–24, 2018.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Mathur, D., Prakash, S., Anand, P., Kaur, H., Agrawal, P., Mehta, A., Kumar, R., Singh, S., and Raghava, G. P. Peplife: a repository of the half-life of peptides. *Scientific reports*, 6(1):36617, 2016.

Michael, R., Bartels, S., González-Duque, M., Zainchkovskyy, Y., Frellsen, J., Hauberg, S., and Boomsma, W. A continuous relaxation for discrete bayesian optimization. *arXiv preprint arXiv:2404.17452*, 2024.

Mohr, S. E., Hu, Y., Ewen-Campen, B., Housden, B. E., Viswanatha, R., and Perrimon, N. Crispr guide rna design for research applications. *The FEBS journal*, 283(17): 3232–3238, 2016.

Naseri, G. and Koffas, M. A. Application of combinatorial optimization strategies in synthetic biology. *Nature communications*, 11(1):2446, 2020.

Nehdi, A., Samman, N., Aguilar-Sánchez, V., Farah, A., Yurdusev, E., Boudjelal, M., and Perreault, J. Novel strategies to optimize the amplification of single-stranded dna. *Frontiers in Bioengineering and Biotechnology*, 8: 401, 2020.

Nisonoff, H., Xiong, J., Allenspach, S., and Listgarten, J. Unlocking guidance for discrete state-space diffusion and flow models. *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rinauro, D. J., Chiti, F., Vendruscolo, M., and Limbocker, R. Misfolded protein oligomers: Mechanisms of formation, cytotoxic effects, and pharmacological approaches against protein misfolding diseases. *Molecular Neurodegeneration*, 19(1):20, 2024.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Schmidt, H., Zhang, M., Chakarov, D., Bansal, V., Mourelatos, H., Sánchez-Rivera, F. J., Lowe, S. W., Ventura, A., Leslie, C. S., and Pritykin, Y. Genome-wide crispr guide rna design and specificity analysis with guidescan2. *Genome biology*, 26(1):1–25, 2025.

Sharma, N., Naorem, L. D., Jain, S., and Raghava, G. P. Toxinpred2: an improved method for predicting toxicity of proteins. *Briefings in bioinformatics*, 23(5):bbac174, 2022.

Sousa, T., Correia, J., Pereira, V., and Rocha, M. Combining multi-objective evolutionary algorithms with deep generative models towards focused molecular design. In *Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24*, pp. 81–96. Springer, 2021.

Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., and Jaakkola, T. Dirichlet flow matching with applications to dna sequence design. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Swanson, K., Walther, P., Leitz, J., Mukherjee, S., Wu, J. C., Shivnaraine, R. V., and Zou, J. Admet-ai: a machine learning admet platform for evaluation of large-scale chemical libraries. *Bioinformatics*, 40(7):btae416, 2024.

Tang, S., Zhang, Y., Tong, A., and Chatterjee, P. Gumbel-softmax flow matching with straight-through guidance for controllable biological sequence generation. *arXiv preprint arXiv:2503.17361*, 2025.

Tominaga, M., Shima, Y., Nozaki, K., Ito, Y., Someda, M., Shoya, Y., Hashii, N., Obata, C., Matsumoto-Kitano, M., Suematsu, K., et al. Designing strong inducible synthetic promoters in yeasts. *Nature Communications*, 15(1):10653, 2024.

Trott, O. and Olson, A. J. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Tsuboyama, K., Dauparas, J., Chen, J., Laine, E., Mohseni Behbahani, Y., Weinstein, J. J., Mangan, N. M., Ovchinnikov, S., and Rocklin, G. J. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.

Ueno, T., Rhone, T. D., Hou, Z., Mizoguchi, T., and Tsuda, K. Combo: An efficient bayesian optimization library for materials science. *Materials discovery*, 4:18–21, 2016.

Yao, Y., Pan, Y., Li, J., Tsang, I., and Yao, X. Proud: Pareto-guided diffusion model for multi-objective generation. *Machine Learning*, 113(9):6511–6538, 2024.

Yuan, Y., Chen, C., Pal, C., and Liu, X. Paretoflow: Guided flows in multi-objective optimization. *arXiv preprint arXiv:2412.03718*, 2024.

Zhang, C., Zhang, X., Freddolino, P. L., and Zhang, Y. Biolip2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, 52(D1):D404–D412, 2024.

Zhang, R., Wu, H., Xiu, Y., Li, K., Chen, N., Wang, Y., Wang, Y., Gao, X., and Zhou, F. Pepland: a large-scale pre-trained peptide representation model for a comprehensive landscape of both canonical and non-canonical amino acids. *arXiv preprint arXiv:2311.04419*, 2023.

Zhou, R., Jiang, Z., Yang, C., Yu, J., Feng, J., Adil, M. A., Deng, D., Zou, W., Zhang, J., Lu, K., et al. All-small-molecule organic solar cells with over 14% efficiency by optimizing hierarchical morphologies. *Nature communications*, 10(1):5393, 2019.

Zitzler, E. and Thiele, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pp. 292–301. Springer, 1998.

Zitzler, E., Laumanns, M., and Thiele, L. Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103, 2001.

A. Weight Vector Generation

To steer the generation towards diverse Pareto-efficient solutions, we introduce a set of weight vectors $\{\omega^k\}_{k=1}^M$ that uniformly cover the N-dimensional Pareto Front. Intuitively, each ω encodes a particular trade-off among the N objectives, so sampling different ω promotes exploration of distinct regions of the Pareto front. Concretely, we construct these vectors via the Das–Dennis simplex lattice with H subdivisions, yielding components

$$\omega_i = \frac{k_i}{H}, \quad k_i \in \mathbb{Z}_{\geq 0}, \quad \sum_{i=1}^{N} k_i = H, \tag{13}$$

and then draw one ω randomly before the following steps. This defines one direction we want to optimize in the state space for the current run.

B. Discrete Flow Matching

In the discrete setting, we consider data $x = (x_1, \ldots, x_d)$ taking values in a finite state space $S = \mathcal{T}^d$, where $\mathcal{T} = [K] = \{1, 2, \ldots, K\}$ is called the vocabulary. We model a continuous-time Markov chain (CTMC) $\{X_t\}_{t\in[0,1]}$ whose time-dependent transition rates $u_t(y, x)$ transport probability mass from an initial distribution p_0 to a target distribution p_1 (Gat et al., 2024). The marginal probability at time t is denoted $p_t(x)$, and its evolution is governed by the Kolmogorov forward equation

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(y) \;=\; \sum_{x\in S} u_t(y, x)\, p_t(x)\,. \tag{14}$$

The learnable velocity field $u_t(y, x)$ is defined as the sum of factorized velocities:

$$u_t(y, x) = \sum_i \delta(y^{\bar{i}}, x^{\bar{i}}) u_t^i(y^i, x), \tag{15}$$

where $\bar{i} = (1, \ldots, i-1, i+1, \ldots, d)$ denotes all indices excluding i. The rate conditions for factorized velocities $u_t^i(y^i, x)$ are required per dimension $i \in [d]$:

$$u_t(y, x) \geq 0 \text{ for all } y^i \neq x^i, \text{ and } \sum_{y^i\in\mathcal{T}} u_t^i(y^i, x) = 0 \text{ for all } x \in S, \tag{16}$$

so that for small $h > 0$, the one-step kernel

$$p_{t+h|t}(y \mid x) = \delta(y, x) + h\, u_t(y, x) + o(h) \tag{17}$$

remains a proper probability mass function.

The goal of training a discrete flow matching model is to learn the velocity field u_t^θ. Representing the marginal velocity u_t^θ in terms of factorized velocities $u_t^{\theta,i}$ enables the following conditional flow matching loss

$$\mathcal{L}_{\mathrm{CDFM}}(\theta) = \mathbb{E}_{t,Z,X_t\sim p_{t|Z}} \sum_i D_{X_t}^i \left(u_t^i(\cdot, X_t \mid Z), u_t^{\theta,i}(\cdot, X_t) \right), \tag{18}$$

where $t \sim \mathcal{U}[0, 1]$, and $u_t^i(\cdot, x \mid z), u_t^{\theta,i}(\cdot, x) \in \mathbb{R}^{\mathcal{T}}$ satisfy the rate conditions. This means that $u_t^i(\cdot, x \mid z), u_t^{\theta,i}(\cdot, x) \in \Omega_{x^i}$ where, for $\alpha \in \mathcal{T}$, we define

$$\Omega_\alpha = \left\{ v \in \mathbb{R}^{\mathcal{T}} \;\middle|\; v(\beta) \geq 0\; \forall\, \beta \in \mathcal{T} \setminus \{\alpha\},\ \text{and } v(\alpha) = -\sum_{\beta\neq\alpha} v(\beta) \right\} \subset \mathbb{R}^{\mathcal{T}}. \tag{19}$$

This is a convex set, and $D_x^i(u, v)$ is a Bregman divergence defined by a convex function $\Phi_x^i : \Omega_{x^i} \to \mathbb{R}$.

In practice, we can further parametrize the velocity field using a mixture path. Specifically, one defines a mixture path with a scheduler $\kappa_t \in [0, 1]$ so that each coordinate X_t^i equals x_0^i or x_1^i with probabilities $1 - \kappa_t$ and κ_t respectively. The mixture marginal velocity is then obtained by averaging the conditional rates over the posterior of (x_0, x_1) given $X_t = x$, yielding

$$u_t^i(y^i, x) = \sum_{x_1^i} \frac{\dot{\kappa}_t}{1 - \kappa_t} \left[\delta(y^i, x_1^i) - \delta(y^i, x^i) \right] p_{1|t}^i(x_1^i \mid x), \tag{20}$$

where $\dot{\kappa}_t$ denotes the time derivative of κ_t. Therefore, the aim of discrete flow matching model training, which is to learn the velocity field $u_t^i(y^i, x)$, now equals to learning the marginal posterior $p_{1|t}^i(x_1^i \mid x)$. In this case, we can set the Bregman divergence to the generalized KL comparing general vectors $u, v \in \mathbb{R}^m_{\geq 0}$,

$$D(u, v) = \sum_j \Big[u_j \log \frac{u_j}{v_j} - u_j + v_j \Big]. \tag{21}$$

For this choice of D, we get

$$D\left(u_t^i(\cdot, x^i \mid x_0, x_1), u_t^{\theta,i}(\cdot, x)\right) = \frac{\dot{\kappa}_t}{1-\kappa_t} \left[(\delta(x_1^i, x^i) - 1) \log p_{1|t}^{\theta,i}(x_1^i \mid x) + \delta(x_1^i, x^i) - p_{1|t}^{\theta,i}(x^i \mid x) \right] \tag{22}$$

which implements the loss (8) when conditioning on $Z = (X_0, X_1)$. The generalized KL loss also provides an evidence lower bound (ELBO) on the likelihood of the target distribution

$$-\log p_1^\theta(x_1) \leq \mathbb{E}_{t, X_0, X_t \sim p_{t|0,1}} \sum_i D\left(u_t^i(\cdot, X_1^i \mid X_0, x_1), u_t^{\theta,i}(\cdot, X_t)\right), \tag{23}$$

where p_1^θ is the marginal generated by the model at time $t = 1$. Therefore, in addition to training, the generalized KL loss can also be used for evaluation.

C. PepDFM and EnhancerDFM Generate Diverse and Biologically Plausible Sequences

To enable the efficient generation of peptide binders, we developed an unconditional peptide generator, **PepDFM**, based on the Discrete Flow Matching (DFM) framework. The model backbone of PepDFM is a U-Net-style convolutional architecture. We trained PepDFM on a custom dataset that includes all peptides from the PepNN and BioLip2 datasets, as well as sequences from the PPIRef dataset with lengths ranging from 6 to 49 amino acids, finally converging to a training loss of 3.3134 and a validation loss of 3.1051 (Abdin et al., 2022; Zhang et al., 2024; Bushuiev et al., 2023). As described in Section B, the low generalized KL loss during evaluation demonstrates the strong performance of PepDFM. We further investigate the diversity and biological plausibility of peptides generated by PepDFM. Specifically, PepDFM generates peptides with substantially high Hamming distances from the test set, indicating a great degree of diversity and novelty in the generated sequences (Figure 3). Additionally, the Shannon entropy of the generated peptides closely matches that of the test set, highlighting the model's capability to produce biologically plausible peptides with diverse sequence lengths (Figure 3).

EnhancerDFM adopts the same model backbone and melanoma enhancer dataset used in Enhancer DNA design task from Stark, et al. (Stark et al., 2024). We employed the Fréchet Biological distance (FBD) metric from (Stark et al., 2024) to evaluate the performance of EnhancerDFM (Table 2). Specifically, using the same number of function evaluations (NFE), EnhancerDFM achieved a comparable FBD of 5.9 compared with Dirichlet FM of 5.3, significantly lower than the FBD of random sequences, demonstrating EnhancerDFM's ability to design biologically plausible enhancer DNA sequences. Significantly, the best EnhancerDFM model is achieved within 20 training epochs, while the best EnhancerDFM is obtained only in around 1400 training epochs, highlighting discrete flow matching models' superior capability of capturing the underlying data distribution.

D. MOG-DFM Effectively Balances Each Objective Trade-off

To validate that MOG-DFM framework can balance the trade-offs between each objective, we performed two sets of experiments for peptide binder generation with three property guidance, and in ablation experiment settings, we removed one or more objectives. In the binder design task for target 7LUL (affinity, solubility, hemolysis guidance; Table 7), omitting any single guidance causes a collapse in that property, while the remaining guided metrics may modestly improve. Likewise, in the binder design task for target CLK1 (affinity, non-fouling, half-life guidance; Table 8), disabling non-fouling guidance allows half-life to exceed 80 hours but drives non-fouling near zero, and disabling half-life guidance preserves non-fouling yet reduces half-life below 2 hours. In contrast, enabling all guidance signals produces the most balanced profiles across all objectives. These results confirm that MOG-DFM precisely targets chosen objectives while preserving the flexibility to navigate conflicting requirements and push samples toward the Pareto front, thereby demonstrating the correctness and precision of our multi-objective sampling framework.

E. Orthogonal Models Confirm MOG-DFM's Effectiveness

In Section H, we demonstrate the reliability of our score models. We now use external evaluation tools to further confirm that MOG-DFM-designed binders possess desired properties. The average solubility and half-life for each target across all 100 designed peptides were predicted using ADMET-AI (Table 5) (Swanson et al., 2024). ADMET-AI, trained on a different dataset from our solubility and half-life prediction models, predicts average LogS values around –2.5 log mol·L^{-1}, which is well above the conventional –4 threshold for good solubility, and confirms long half-life estimates ($>$ 15 h). These results from an orthogonal predictive model demonstrate MOG-DFM's capability to generate candidates with multiple desirable drug properties.

F. MOG-DFM Outperforms Classical Evolutionary Algorithms

We benchmarked MOG-DFM against four multi-objective optimizers—NSGA-III (Deb & Jain, 2013), SMS-EMOA (Beume et al., 2007), SPEA2 (Zitzler et al., 2001), and MOPSO (Coello & Lechuga, 2002)—on two protein targets: 1B8Q (a small protein with known peptide binders) and PPP5 (a larger protein lacking characterized binders) (Table 4). For each method, we generated 100 peptide binders per target of a specified length, guided by five property objectives (hemolysis, non-fouling, solubility, half-life, and binding affinity), and recorded both the average generation time for one sequence and the mean property scores. Although MOG-DFM requires longer runtimes, it consistently produces the most favorable trade-offs: reducing predicted hemolysis by more than 10%, boosting non-fouling and solubility by approximately 30-50%, and extending half-life by a factor of 3 to 4 compared to the next-best method, while maintaining competitive affinity values. These results demonstrate MOG-DFM's effectiveness in navigating high-dimensional property landscapes to generate peptide binders with well-balanced, optimized profiles. We did not benchmark against ParetoFlow, another multi-objective optimization algorithm that uses flow matching, because it requires score models to take continuous inputs, which is not suitable for our task.

G. Base Model Details

G.1. PepDFM

Model Architecture. The base model is a time-dependent architecture based on U-Net (Ronneberger et al., 2015). It uses two separate embedding layers for sequence and time, followed by five convolutional blocks with varying dilation rates to capture temporal dependencies, while incorporating time-conditioning through dense layers. The final output layer generates logits for each token. We used a polynomial convex schedule with a polynomial exponent of 2.0 for the mixture discrete probability path in the discrete flow matching.

Dataset Curation. The dataset for PepDFM training was curated from the PepNN, BioLip2, and PPIRef dataset (Abdin et al., 2022; Zhang et al., 2024; Bushuiev et al., 2023). All peptides from PepNN and BioLip2 were included, along with sequences from PPIRef ranging from 6 to 49 amino acids in length. The dataset was divided into training, validation, and test sets at an 80/10/10 ratio.

Training Strategy. The training is conducted on a 2xH100 NVIDIA NVL GPU system with 94 GB of VRAM for 200 epochs with batch size 512. The model checkpoint with the lowest evaluation loss was saved. The Adam optimizer was employed with a learning rate 1e-4. A learning rate scheduler with 20 warm-up epochs and cosine decay was used, with initial and minimum learning rates both 1e-5. The embedding dimension and hidden dimension were set to be 512 and 256 respectively for the base model.

Dynamic Batching. To enhance computational efficiency and manage variable-length token sequences, we implemented dynamic batching. Drawing inspiration from ESM-2's approach (Lin et al., 2023), input peptide sequences were sorted by length to optimize GPU memory utilization, with a maximum token size of 100 per GPU.

G.2. EnhancerDFM

Model Architecture. The base model for EnhancerDFM applies the same architecture as the PepDFM. We also used a polynomial convex schedule with a polynomial exponent of 2.0 for the mixture discrete probability path in the discrete flow matching.

Dataset Curation. The dataset for EnhancerDFM training is curated by (Stark et al., 2024). The dataset contains 89k

enhancer sequences from human melanoma cells (Atak et al., 2021). Each sequence is of length 500 paired with cell class labels determined from ATAC-seq data (Buenrostro et al., 2013). There are 47 such classes of cells in total, with details displayed in Table 11 (Atak et al., 2021). We applied the same dataset split strategy as (Stark et al., 2024).

Training Strategy. The training is conducted on a 2xH100 NVIDIA NVL GPU system with 94 GB of VRAM for 1500 epochs with batch size 256. The model checkpoint with the lowest evaluation loss was saved. The Adam optimizer was employed with a learning rate 1e-3. A learning rate scheduler with 150 warm-up epochs and cosine decay was used, with initial and minimum learning rates both 1e-4. Both the embedding dimension and hidden dimension were set to be 256 for the base model.

H. Score Model Details

We collected hemolysis (9,316), non-fouling (17,185), solubility (18,453), and binding affinity (1,781) data for classifier training from the PepLand and PeptideBERT datasets (Zhang et al., 2023; Guntuboina et al., 2023). All sequences taken are wild-type L-amino acids and are tokenized and represented by ESM-2 protein language model (Lin et al., 2023).

H.1. Boosted Trees for Classification

For hemolysis, non-fouling, and solubility classification, we trained XGBoost boosted tree models for logistic regression. We split the data into 0.8/0.2 train/validation using stratified splits from scikit-learn (Pedregosa et al., 2011) and generated mean pooled ESM-2-650M (Lin et al., 2023) embeddings as input features to the model. We ran 50 trials of OPTUNA (Akiba et al., 2019) search to determine the optimal XGBoost hyperparameters (Table 1), tracking the best binary classification F1 scores. The best models for each property reached F1 scores of: 0.58, 0.71, and 0.68 on the validation sets accordingly.

Table 1. XGBoost Hyperparameters for Classification

Hyperparameter	**Value/Range**
Objective	`binary:logistic`
Lambda	$[1e{-}8, 10.0]$
Alpha	$[1e{-}8, 10.0]$
Colsample by Tree	$[0.1, 1.0]$
Subsample	$[0.1, 1.0]$
Learning Rate	$[0.01, 0.3]$
Max Depth	$[2, 30]$
Min Child Weight	$[1, 20]$
Tree Method	`hist`

H.2. Binding Affinity Score Model

We developed an unpooled reciprocal attention transformer model to predict protein-peptide binding affinity, leveraging latent representations from the ESM-2 650M protein language model (Lin et al., 2023). Instead of relying on pooled representations, the model retains unpooled token-level embeddings from ESM-2, which are passed through convolutional layers followed by cross-attention layers. The binding affinity data was split into a 0.8/0.2 ratio, maintaining similar affinity score distributions across splits. We used OPTUNA (Akiba et al., 2019) for hyperparameter optimization tracing validation correlation scores. The final model was trained for 50 epochs with a learning rate of 3.84e-5, a dropout rate of 0.15, 3 initial CNN kernel layers (dimension 384), 4 cross-attention layers (dimension 2048), and a shared prediction head (dimension 1024) in the end. The classifier reached 0.64 Spearman's correlation score on validation data.

H.3. Half-Life Score Model

Dataset Curation. The half-life dataset is curated from three publicly available datasets: PEPLife, PepTherDia, and THPdb2 (Mathur et al., 2016; D'Aloisio et al., 2021; Jain et al., 2024). Data related to human subjects were selected, and entries with missing half-life values were excluded. After removing duplicates, the final dataset consists of 105 entries.

Pre-training on stability data. Given the small size of the half-life dataset, which is insufficient for training a model to

capture the underlying data distribution, we first pre-trained a score model on a larger stability dataset to predict peptide stability (Tsuboyama et al., 2023). The model consists of three linear layers with ReLU activation functions, and a dropout rate of 0.3 was applied. The model was trained on a 2xH100 NVIDIA NVL GPU system with 94 GB of VRAM for 50 epochs. The Adam optimizer was employed with a learning rate 1e-2. A learning rate scheduler with 5 warm-up epochs and cosine decay was used, with initial and minimum learning rates both 1e-3. After training, the model achieved a validation Spearman's correlation of 0.7915 and an R^2 value of 0.6864, demonstrating the reliability of the stability score model.

Fine-tuning on half-life data. The pre-trained stability score model was subsequently fine-tuned on the half-life dataset. Since half-life values span a wide range, the model was adapted to predict the base-10 logarithm of the half-life (h) values to stabilize the learning process. After fine-tuning, the model achieved a validation Spearman's correlation of 0.8581 and an R^2 value of 0.5977.

I. Sampling Details

I.1. Peptide Binder Generation Tasks

Score Model Settings. To align all objectives as maximization, we convert the predicted hemolysis rate h into a score $1 - h$, so that lower hemolysis yields a higher value. We also cap the predicted log-scale half-life at 2 (i.e., 100 h) to prevent it from dominating the optimization and ensure balanced trade-offs across all properties. For the remaining objectives—non-fouling, solubility, and binding affinity—we directly employ their model outputs during sampling.

Hyperparameter Settings. The hyperparameters were set as follows: The number of divisions used in generating weight vectors, num_div, was set to 64, λ to 1.0, β to 1.0, α_r to 0.5, τ to 0.3, η to 1.0, Φ_{init} to $45°$, Φ_{min} to $15°$, Φ_{max} to $75°$. The total sampling step T was 100.

Importance Vectors. In the task with five property guidance, the importance vector was set to $[1, 1, 1, 0.5, 0.2]$, each corresponding to hemolysis, non-fouling, solubility, half-life, and binding affinity guidance, respectively. For the two tasks with only three property guidance, the importance vector was set to $[1, 1, 0.1]$ for solubility, hemolysis, and binding affinity guidance, respectively, and $[1, 0.5, 0.2]$ for non-fouling, half-life, and binding affinity guidance, respectively. The rationale for setting the importance values is based on the range lengths of the properties: hemolysis, non-fouling, and solubility each have a range length of 1.0, half-life has a range length of 2.0, and binding affinity has a range length of 10.0. The importance values were assigned inversely proportional to these range lengths.

I.2. Enhancer DNA Generation Tasks.

Hyperparameter Settings. The hyperparameters were set the same as those in peptide binder generation tasks, except that the total sampling step T was set to 800.

Importance Vectors. The importance vector was set to be [1, 10] for the first task and [1, 100] for the second task, with the first value corresponding to the enhancer class guidance and the second value corresponding to the DNA shape guidance. The rationale for assigning these importance values is based on the range lengths of the properties: enhancer class probability has a range length of 1.0, HelT shape values have a range length of 2.0, and Rise shape values have a range length of 0.1. The importance values were assigned inversely proportional to these range lengths.

J. Hyperparameter Sensitivity Benchmark

There are several hyperparameters in MOG-DFM whose settings may affect generative performance. To assess this sensitivity, we evaluated peptide binder design across a broad range of values for each parameter (Table 10). We find that increasing the number of sampling steps consistently improves all performance metrics, as finer discretization more closely approximates the continuous-time dynamics. In contrast, setting the initial hypercone angle Φ_{init} too small or too large both degrade results: an overly narrow cone restricts exploration, while an overly wide cone dilutes directional guidance. The importance weights also play a critical role in balancing multiple objectives. Because each property can vary over a different numerical range, we initialize each weight inversely proportional to the maximum observed improvement of that property, thereby normalizing all guidance signals to roughly unit scale. This allows for similar improvements for each objective, otherwise, the improvements for some objectives may stagnate. By comparison, the remaining hyperparameters (i.e. β, λ, α_r, η, τ, and the bounds $\Phi_{\text{min}}, \Phi_{\text{max}}$) exhibit only modest impact on outcomes, indicating that MOG-DFM is robust to moderate variations in these settings.

K. Adaptive Hypercone Filtering Enhances Multi-Objective Optimization

To quantify the contribution of our adaptive hypercone mechanism, we performed an ablation study on three protein targets (3IDJ, 4E-BP2, and EWS::FLI1), generating 100 peptide binders for each target (Table 6). Removing hypercone filtering entirely ("w/o filtering") causes a dramatic collapse in half-life—from roughly 30–35 h down to 4–13 h—while leaving non-fouling and solubility largely unchanged, indicating that filtering out poorly aligned moves is essential for optimizing objectives that require gradual, coordinated changes. Introducing static hypercone gating without angle adaptation ("w/o adaptation") recovers much of the half-life gains (to 23–37 h), but at the expense of reduced non-fouling and solubility scores and only marginal improvements in affinity. In contrast, the full MOG-DFM—with both directional hypercone filtering and adaptive angle updates—simultaneously elevates half-life and maintains strong performance across all five objectives. This effect is especially pronounced on disordered targets (4E-BP2 and EWS::FLI1), where dynamic cone adjustment is essential for navigating the irregular, non-convex Pareto landscapes.

L. Additional Proof

Claim: MOG-DFM directs the discrete generation process toward the Pareto front by inducing a positive expected improvement in the direction of a specified weight vector $\omega \in \mathbb{R}^N$.

Proof: Let $\mathcal{S} = \mathcal{T}^d$ be the discrete sequence space over vocabulary $\mathcal{T}$, and let $x \in \mathcal{S}$ denote the current sequence state at time $t \in [0, 1]$. Assume the multi-objective score function $s : \mathcal{S} \to \mathbb{R}^N$ is measurable, with N scalar objectives. Define the improvement vector at a candidate transition $y^i \in \mathcal{T} \setminus \{x^i\}$ at position $i \in \{1, \ldots, d\}$ as:

$$\Delta s(y^i, x) := s(x^{(i \to y^i)}) - s(x),$$

where $x^{(i \to y^i)}$ denotes the sequence x with token x^i replaced by y^i.

Let $\omega \in \mathbb{R}^N$ be a fixed unit-norm trade-off vector sampled uniformly from the Das–Dennis lattice covering the simplex Δ^{N-1}. Define the directional improvement of a transition y^i as:

$$D(y^i, x; \omega) := \Delta s(y^i, x) \cdot \omega.$$

Define the set of feasible transitions (those within the hypercone of angle $\Phi \in (0, \pi)$) at time t as:

$$Y^i(x, \omega, \Phi) := \left\{ y^i \in \mathcal{T} \setminus \{x^i\} \;\middle|\; \arccos\left(\frac{\Delta s(y^i, x) \cdot \omega}{\|\Delta s(y^i, x)\| \cdot \|\omega\|} \right) \leq \Phi \right\}.$$

Let $\mu_t^i(\cdot \mid x, \omega)$ be the conditional probability measure over feasible transitions defined by:

$$\mu_t^i(y^i \mid x, \omega) := \frac{\exp\big(\Delta S(y^i, x, \omega)\big)}{Z(x, \omega)} \cdot \mathbf{1}_{\{y^i \in Y^i(x, \omega, \Phi)\}},$$

where $\Delta S(\cdot)$ is the rank-directional guidance score and $Z(x, \omega) := \sum_{y^i \in Y^i} \exp\big(\Delta S(y^i, x, \omega)\big)$ is the normalizing partition function. Assume that $Y^i(x, \omega, \Phi)$ is non-empty, or else the algorithm falls back to selecting the best y^i with $D(y^i, x; \omega) > 0$ by construction.

We now consider the expected improvement in the direction of ω over all guided transitions:

$$\mathbb{E}_{i \sim \mathcal{U}[d],\, y^i \sim \mu_t^i(\cdot \mid x, \omega)} \big[D(y^i, x; \omega)\big] = \frac{1}{d} \sum_{i=1}^{d} \sum_{y^i \in Y^i(x, \omega, \Phi)} D(y^i, x; \omega) \cdot \mu_t^i(y^i \mid x, \omega).$$

Since each $y^i \in Y^i(x, \omega, \Phi)$ satisfies $\arccos\left(\frac{\Delta s(y^i, x) \cdot \omega}{\|\Delta s(y^i, x)\| \cdot \|\omega\|}\right) \leq \Phi < \pi$, it follows that $D(y^i, x; \omega) > 0$ for all $y^i \in Y^i$. Moreover, $\mu_t^i(y^i \mid x, \omega) > 0$ by construction.

Therefore, each term in the sum is strictly positive, and thus:

$$\mathbb{E}[\Delta s(x_{\text{new}}, x) \cdot \omega] > 0,$$

Table 2. **Evaluation of unconditional EnhancerDNA generation.** Each method generates 10k sequences, and we compare their empirical distributions with the data distributions using the Fréchet Biological distance (FBD) metric. NFE refers to number of function evaluations. # Tranining Epochs refers to the number of training epochs needed to get the model checkpoint for this evaluation. The Random Sequence baseline shows the FBD for the same number and length of sequences with uniform randomly chosen nucleotides. Dirichlet FM refers to the Dirichlet Flow Matching model.

	FBD	**NFE**	**# Training Epochs**
Random Sequence	622.8	-	-
Dirichlet FM	5.3	100	1400
EnhancerDFM	5.9	100	20

Table 3. MOG-DFM generates peptide binders for 10 diverse protein targets, optimizing five therapeutic properties: hemolysis, non-fouling, solubility, half-life (in hours), and binding affinity. Each value represents the average of 100 MOG-DFM-designed binders.

Target	**Binder Length**	**Hemolysis (↓)**	**Non-Fouling (↑)**	**Solubility (↑)**	**Half-Life (↑)**	**Affinity (↑)**
AMHR2	8	0.0755	0.8352	0.8219	31.624	7.3789
AMHR2	12	0.0570	0.8419	0.8279	28.761	7.4274
AMHR2	16	0.0618	0.7782	0.7428	31.227	7.6099
EWS::FLI1	8	0.0809	0.8508	0.8296	47.169	6.2251
EWS::FLI1	12	0.0616	0.8302	0.8130	34.225	6.3631
EWS::FLI1	16	0.0709	0.7787	0.7400	34.192	6.5912
MYC	8	0.0809	0.8135	0.8005	39.836	6.8488
OX1R	10	0.0741	0.8115	0.7969	33.533	7.4162
DUSP12	9	0.0735	0.8360	0.8216	33.754	6.4946
1B8Q	8	0.0744	0.8334	0.827	33.243	5.932
1E6I	6	0.0887	0.7884	0.7793	41.164	4.9621
3IDJ	7	0.0924	0.8246	0.7992	30.388	7.6304
5AZ8	11	0.0698	0.8462	0.8420	28.726	6.6051
7JVS	11	0.0628	0.8390	0.8206	32.834	6.9569

where $x_{\text{new}} = x^{(i \to y^i)}$ is the updated sequence following a guided and filtered transition.

Hence, the MOG-DFM procedure ensures that in expectation, the sampling dynamics induce forward motion along the Pareto trade-off direction ω, thereby steering generation toward the Pareto frontier.

□

Table 4. MOG-DFM outperforms traditional multi-objective optimization algorithms in designing peptide binders guided by five objectives. Each value represents the average of 100 designed binders. The table also records the average runtime for each algorithm to design a single binder. The best result for each metric is highlighted in bold.

Target	Method	Time (s)	Hemolysis (↓)	Non-Fouling	Solubility	Half-Life	Affinity
1B8Q	MOPSO	8.54	0.1066	0.4763	0.4684	4.449	6.0594
	NSGA-III	33.13	0.0862	0.5715	0.5825	7.324	7.2178
	SMS-EMOA	**8.21**	0.1196	0.3450	0.3511	3.023	5.955
	SPEA2	17.48	0.0819	0.4973	0.5057	4.126	**7.324**
	MOG-DFM	43.00	**0.0785**	**0.8445**	**0.8455**	**27.227**	5.9094
PPP5	MOPSO	11.34	0.0883	0.4711	0.4255	1.769	6.6958
	NSGA-III	37.30	**0.0479**	0.7138	0.7066	2.901	7.3789
	SMS-EMOA	**8.43**	0.1242	0.4269	0.4334	1.031	6.2854
	SPEA2	19.02	0.0555	0.6221	0.6098	2.613	**7.6253**
	MOG-DFM	90.00	0.0617	**0.7738**	**0.751**	**27.775**	6.8197

Table 5. Average solubility (LogS) and half-life (in hours) metrics computed by ADMET-AI for each target across the 100 MOG-DFM-designed binders.

Target	LogS	Half-Life
AMHR2	-2.3931	15.505
AMHR2	-2.5055	18.777
AMHR2	-2.5784	16.463
EWS::FLI1	-2.3869	18.945
EWS::FLI1	-2.3813	16.305
EWS::FLI1	-2.5457	15.984
MYC	-2.4053	16.491
OX1R	-2.4772	23.002
DUSP12	-2.4333	19.258
1B8Q	-2.3203	18.7862
1E6I	-2.0394	19.9358
3IDJ	-2.4193	20.3586
5AZ8	-2.5964	16.3016
7JVS	-2.4824	20.2565

Table 6. **Ablation study results for the adaptive hypercone filtering module in MOG-DFM.** Three settings are evaluated: 'w/o filtering' indicates the module is completely disabled, 'w/o adaptation' means the module is enabled but the hypercone is not adaptive, and 'MOG-DFM' represents the complete algorithm. For each setting, 100 peptide binders were designed, with lengths of 7, 12, and 12 for the targets 3IDJ, 4E-BP2, and EWS::FLI1, respectively.

Target	Method	Hemolysis (↓)	Non-Fouling	Solubility	Half-Life	Affinity
3IDJ	w/o filtering	0.0660	0.8430	0.8482	12.50	7.3730
	w/o adaptation	0.0856	0.8060	0.7970	37.17	7.3142
	MOG-DFM	0.0924	0.8246	0.7992	30.39	7.6304
4E-BP2	w/o filtering	0.0504	0.8582	0.8600	12.62	6.5066
	w/o adaptation	0.0638	0.8418	0.8234	23.44	6.4548
	MOG-DFM	0.0698	0.8210	0.8050	34.88	6.5824
EWS::FLI1	w/o filtering	0.0450	0.8596	0.8570	4.40	6.1392
	w/o adaptation	0.0620	0.8444	0.8482	28.82	6.2118
	MOG-DFM	0.0616	0.8302	0.8130	34.225	6.3631

Table 7. Ablation results for peptide binder design targeting PDB 7LUL with different guidance settings. For each setting, 100 binders of length 7 were designed.

Guidance Settings			**Affinity**	**Solubility**	**Hemolysis (↓)**
Affinity	**Solubility**	**Hemolysis**			
✓	✓	×	6.3489	0.8890	0.0620
×	✓	✓	5.0514	0.9482	0.0406
✓	×	✓	6.9060	0.4224	0.0488
✓	✓	×	6.5304	0.8975	0.1019
×	×	✓	5.0761	0.7148	0.0163
×	✓	×	5.2434	0.9772	0.0955
✓	×	×	7.4834	0.1218	0.3281
×	×	×	5.5631	0.3736	0.1567

Table 8. Ablation results for peptide binder design targeting PDB CLK1 with different guidance settings. For each setting, 100 binders of length 12 were designed.

Guidance Settings			**Affinity**	**Non-Fouling**	**Half-Life**
Affinity	**Non-Fouling**	**Half-Life**			
✓	✓	✓	6.9194	0.7401	51.73
×	✓	✓	6.4735	0.8107	60.75
✓	×	✓	7.5360	0.3062	84.70
✓	✓	×	7.4150	0.8560	1.24
×	×	✓	6.2363	0.2624	96.44
×	✓	×	6.1378	0.9503	0.94
✓	×	×	8.5943	0.2439	3.15
×	×	×	5.8926	0.3999	1.94

Table 9. **Performance evaluation of MOG-DFM in guided DNA sequence generation.** Task 1 guides the generation towards the HelT shape and enhancer class 1, while Task 2 targets the Rise shape and enhancer class 16. The table presents the predicted DNA shape values (HelT for Task 1, Rise for Task 2) and enhancer class probabilities (class 1 for Task 1, class 16 for Task 2) under various guidance conditions. The 'Shape' column shows the predicted DNA shape values obtained using Deep DNAshape, and the 'Class Prob' column displays the predicted enhancer class probabilities. Ablation studies were conducted by removing one or both guidance criteria, as shown by the rows corresponding to different combinations of shape and class guidance. For each setting, 5 enhancer DNA sequences were designed.

Guidance Settings		**Task 1**		**Task 2**	
Shape	**Class**	**Class Prob**	**Shape**	**Class Prob**	**Shape**
		0.7504	36.0100	0.9960	3.3640
		0.6507	36.0100	0.9922	3.3680
✓	✓	0.6821	36.0000	0.9864	3.3669
		0.7097	36.0000	0.9976	3.3680
		0.6425	36.0000	0.9961	3.3623
		0.9999	34.3274	1.0000	3.3368
		0.9999	34.4715	1.0000	3.3345
✓	×	0.9989	34.4257	0.9999	3.3348
		0.9997	34.5226	0.9994	3.3357
		0.9998	34.4210	1.0000	3.3340
		0.0026	36.0017	2.36E-05	3.3690
		0.0055	36.0238	0.0005	3.3647
×	✓	0.0062	36.0214	0.0114	3.3705
		0.0186	36.0396	0.0001	3.3717
		0.0051	36.0304	0.0054	3.3669
		0.0362	34.7379	0.0008	3.3283
		0.0364	34.5350	0.0057	3.3258
×	×	0.0309	34.5720	0.0476	3.3268
		0.0138	34.3060	0.0632	3.3378
		0.0213	34.5500	0.0003	3.3320

Table 10. Hyperparameter sensitivity benchmark for MOG-DFM in peptide binder generation, guided by five objectives. For each setting, 100 peptide binders are designed with a length matching that of the pre-existing binder for each target.

Hyper parameter	Target	Value	Hemolysis (↓)	Non-Fouling	Solubility	Half-Life	Affinity
num_div	6MLC	32	0.0994	0.8088	0.7924	38.39	6.5436
		64	0.0863	0.8280	0.8232	34.91	6.3260
		128	0.0890	0.8438	0.8386	32.97	6.4197
β	4IU7	0.5	0.0829	0.7894	0.761	28.10	6.7884
		1	0.0684	0.8388	0.8321	41.78	7.0002
		1.5	0.0585	0.8588	0.8582	47.65	7.0505
		2	0.0615	0.8461	0.8416	53.45	7.0169
λ	1AYC	0.5	0.0703	0.8168	0.8152	30.89	6.4838
		1	0.0647	0.8362	0.8207	33.28	6.4549
		2	0.0587	0.8690	0.8461	41.90	6.5317
α_r	2Q8Y	0.1	0.0777	0.8361	0.8051	37.83	6.0569
		0.3	0.0718	0.8441	0.8280	38.83	6.0484
		0.5	0.0718	0.8529	0.8421	31.45	6.0445
		0.7	0.0688	0.8403	0.8377	35.50	6.0839
		0.9	0.0813	0.8288	0.8091	45.25	6.1599
η	2LTV	0.5	0.0633	0.8437	0.8368	29.48	7.3657
		1	0.0601	0.8256	0.8144	24.47	7.3111
		2	0.0624	0.8125	0.7887	35.13	7.1974
Φ_{init}	5M02	15	0.0746	0.8285	0.8007	34.04	7.0335
		30	0.0792	0.8393	0.8187	35.60	7.0251
		45	0.0747	0.8338	0.8192	36.29	7.0944
		60	0.0813	0.8095	0.7970	38.25	7.0932
		75	0.0830	0.8139	0.7949	33.29	7.1261
$[\Phi_{\min}, \Phi_{\max}]$	3EQS	[0,90]	0.0572	0.8385	0.8200	26.64	8.2201
		[15,75]	0.0599	0.8373	0.8116	29.56	8.1673
		[30,60]	0.0614	0.8159	0.8020	35.71	8.2313
τ	5E1C	0	0.0614	0.8252	0.8119	24.57	7.0112
		0.1	0.0650	0.8017	0.7835	31.19	7.1067
		0.3	0.0595	0.8224	0.8088	28.72	7.0756
		0.5	0.0555	0.8310	0.8043	24.03	7.0862
		0.7	0.0590	0.8360	0.8078	28.27	7.0477
T	5KRI	50	0.0757	0.7386	0.7219	15.22	6.9155
		100	0.0580	0.8617	0.8504	30.25	6.9946
		200	0.0525	0.8695	0.8621	41.53	7.2166
		500	0.0518	0.8799	0.8760	57.65	7.2172
importance weights	4EZN	[1,1,1,0.5,0.2]	0.0877	0.5735	0.5485	28.17	6.4190
		[1,1,1,0.5,0.1]	0.0836	0.6003	0.5738	21.99	6.3409
		[1,1,1,1,0.1]	0.0892	0.5549	0.5272	33.58	6.3844
		[1,1,1,1,0.2]	0.0958	0.5939	0.5647	34.80	6.4281
		[1,1,1,1,1]	0.0960	0.5377	0.5007	29.65	6.8613

Table 11. Motif clusters and associated properties of enhancer DNA sequences. In this paper, each class refers to its corresponding cluster ID.

Cluster ID	# of explainable ASCAVs	Motif Annotation	# of Motifs in the cluster
cluster_1	3278	ATF	71
cluster_2	1041	CTCF	85
cluster_3	2480	EBOX	91
cluster_4	4011	AP1	191
cluster_5	1165	RUNX	37
cluster_6	789	SP	20
cluster_7	1285	ETS	33
cluster_8	544	TEAD	9
cluster_9	1024	TFAP	53
cluster_10	334	Other	4
cluster_11	935	SOX	17
cluster_12	1010	CTCFL	16
cluster_13	696	GATA	7
cluster_14	141	Other	2
cluster_15	601	TEAD	6
cluster_16	805	Other	7
cluster_17	270	Other	4
cluster_18	475	Other	5
cluster_19	473	ZNF	6
cluster_20	395	Other	4
cluster_21	393	Other	4
cluster_22	768	NRF	8
cluster_23	214	Other	2
cluster_24	336	Other	2
cluster_25	375	Other	3
cluster_26	215	Other	2
cluster_27	234	Other	2
cluster_28	354	Other	3
cluster_29	210	Other	2
cluster_30	200	Other	2
cluster_31	218	Other	2
cluster_32	415	Other	2
cluster_33	387	SOX	2
cluster_34	116	Other	1
cluster_35	121	Other	1
cluster_36	394	Other	2
cluster_37	112	Other	1
cluster_38	111	Other	1
cluster_39	107	Other	1
cluster_40	118	Other	1
cluster_41	144	Other	1
cluster_42	105	Other	1
cluster_43	102	Other	1
cluster_44	108	Other	1
cluster_45	114	Other	1
cluster_46	118	Other	1
cluster_47	119	Other	1

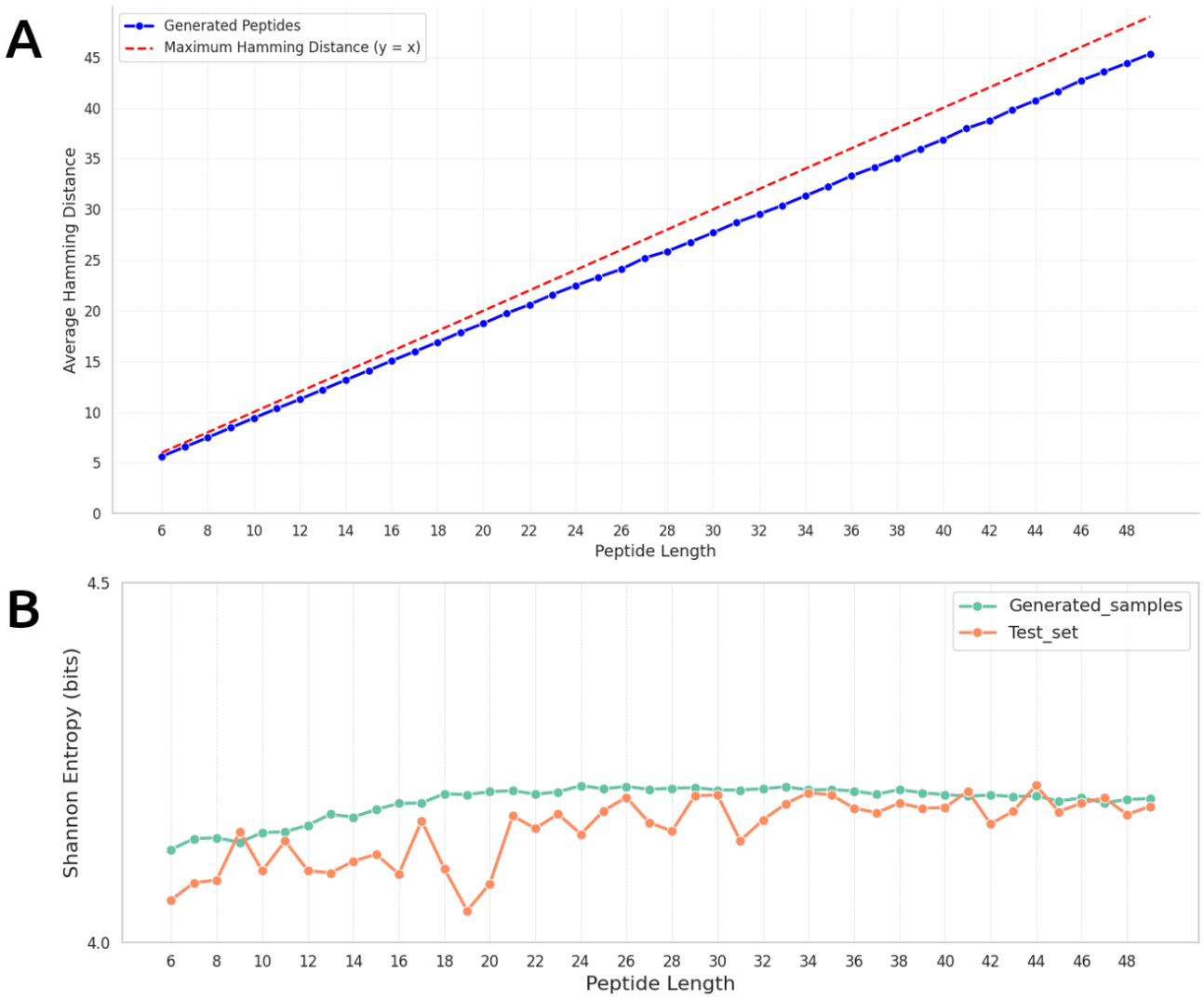

Figure 3. **(A)** The Hamming distance of sampled peptides of different lengths to the peptides of the same length in the test set. **(B)** The Shannon Entropy of sampled peptides of different lengths to the peptides of the same length in the test set.

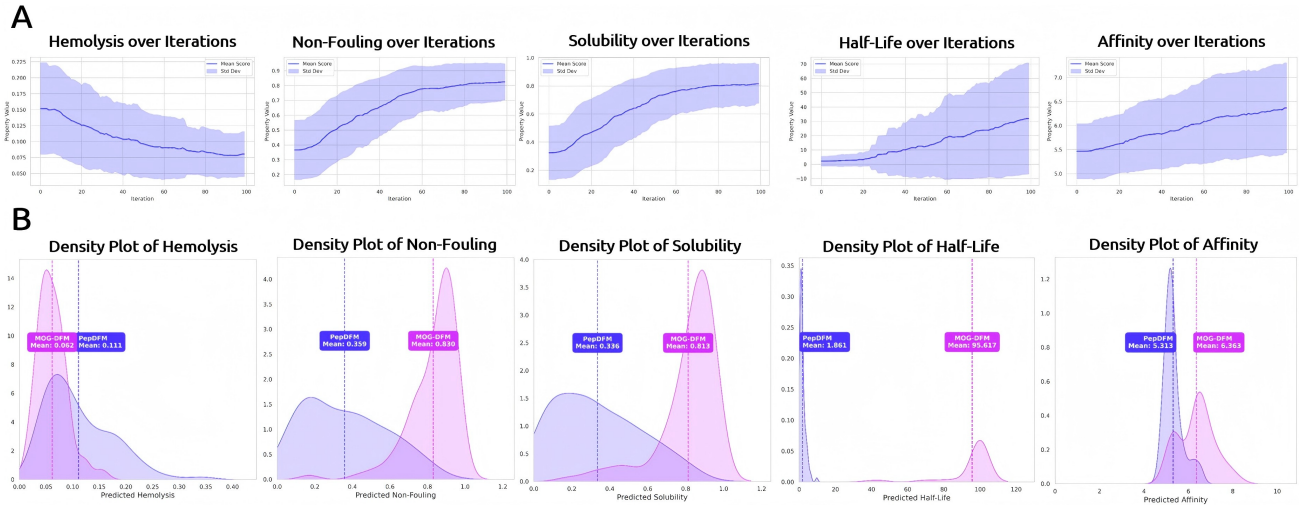

Figure 4. **(A)** Plots showing the mean scores for each property across the number of iterations during MOG-DFM's design of binders of length 12-aa for EWS::FLI1. **(B)** Density plots illustrating the distribution of predicted property scores for MOG-DFM-designed EWS::FLI1 binders of length 12-aa, compared to the peptides generated unconditionally by PepDFM. Please zoom in for better viewing.

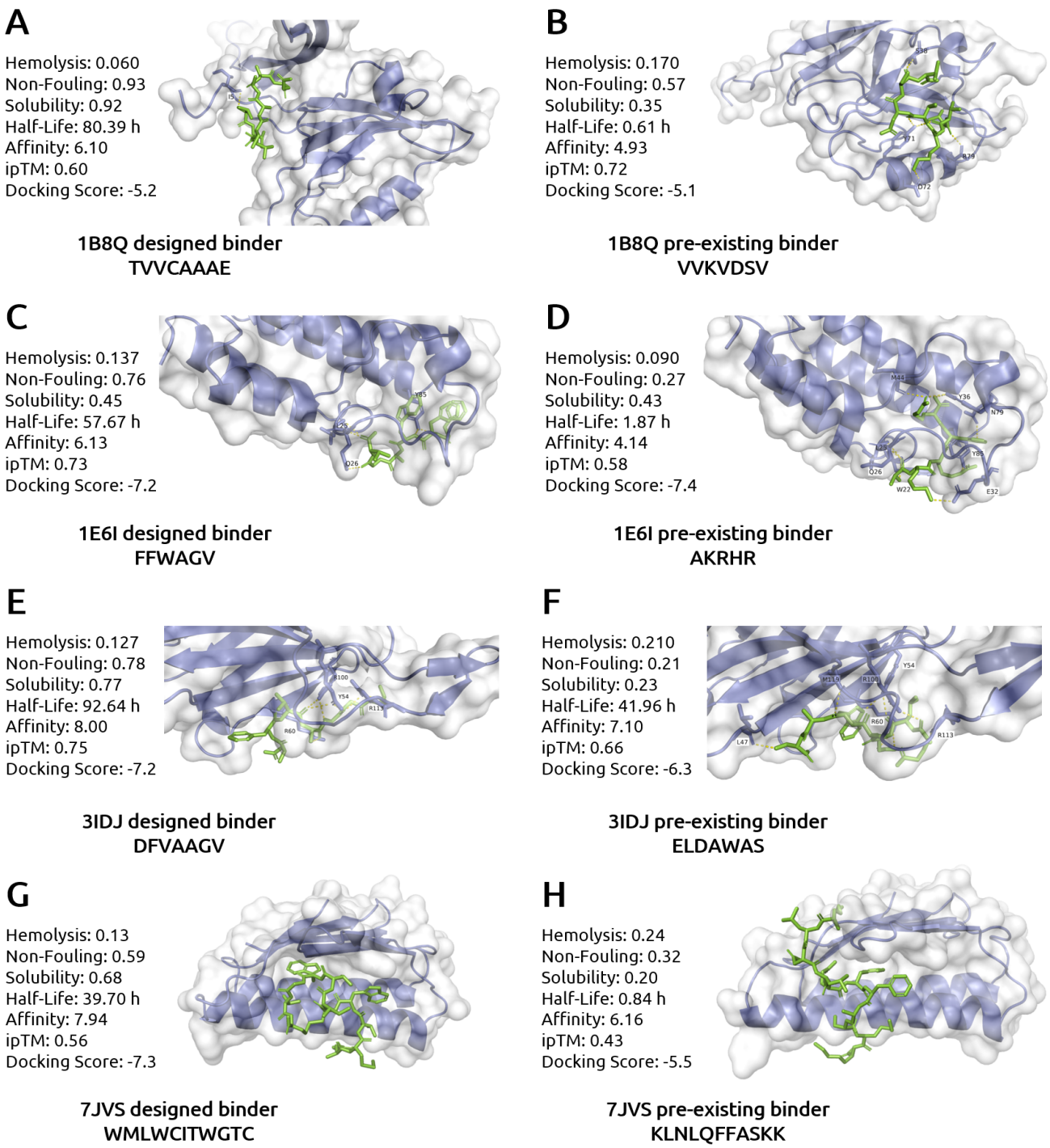

Figure 5. **Complex structures of target proteins with pre-existing binders. (A)-(B)** 1B8Q, **(C)-(D)** 1E6I, **(E)-(F)** 3IDJ, **(G)-(H)** 7JVS. Each panel shows the complex structure of the target with either a MOG-DFM-designed binder or its pre-existing binder. For each binder, five property scores are provided, as well as the ipTM score from AlphaFold3 and the docking score from AutoDock VINA. Interacting residues on the target are visualized.

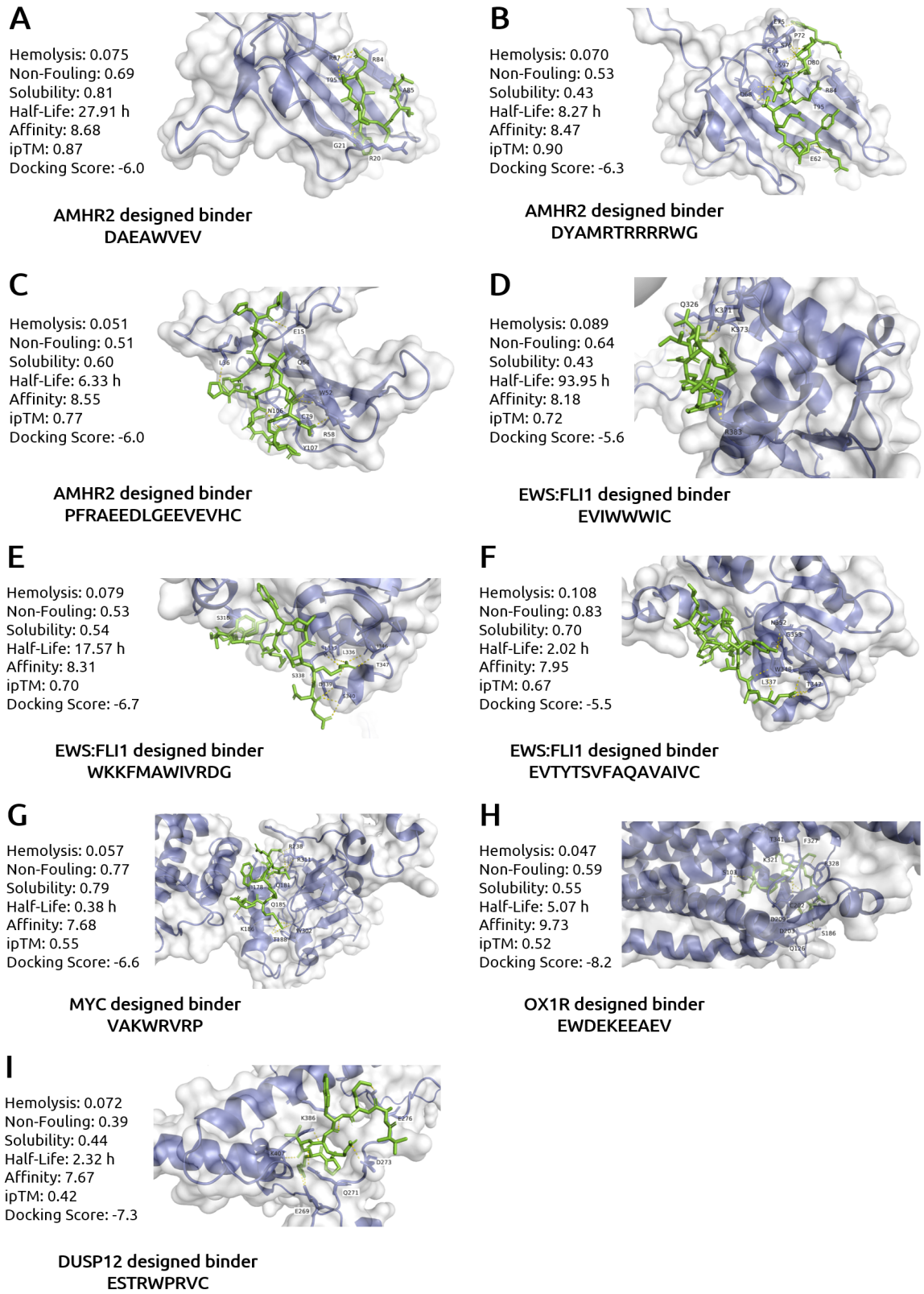

Figure 6. **Complex structures of target proteins without pre-existing binders.** **(A)-(C)** AMHR2, **(D)-(F)** EWS::FLI1, **(G)** MYC, **(H)** OX1R, **(I)** DUSP12. Each panel shows the complex structure of the target with a MOG-DFM-designed binder. For each binder, five property scores are provided, as well as the ipTM score from AlphaFold3 and the docking score from AutoDock VINA. Interacting residues on the target are visualized.

Algorithm 1 MOG-DFM: Multi-Objective-Guided Discrete Flow Matching

Input: Pre-trained discrete flow matching model, multi-objective score functions
Output: Sequence x_1 with multi-objective optimized properties
Initialize:
  Sample an initial sequence x_0 uniformly from the discrete state space S
  Generate a set of weight vectors $\{\omega_k\}_{k=1}^{M}$ that uniformly cover the N-dimensional Pareto front
  Select a weight vector ω randomly from $\{\omega_k\}$
for $t = 0$ to 1 with step size $h = \frac{1}{T}$ **do**
  Step 1: Guided Transition Scoring
    Select a position i in the sequence to update
    For each candidate transition $y_i \neq x_i$:
      Compute the normalized rank score $I_n(y_i, x)$ for each objective n
      Compute $D(y_i, x, \omega)$ based on the alignment of improvements with ω
      Combine rank and direction components:

$$\Delta S(y_i, x, \omega) = \text{Norm}\left[\frac{1}{N}\sum_{n=1}^{N} I_n(y_i, x)\right] + \lambda \cdot \text{Norm}\left[D(y_i, x, \omega)\right]$$

    Re-weight the original velocity field $u_i(y_i, x)$ by the combined score
  Step 2: Adaptive Hypercone Filtering
    Compute angle α_i between improvement vector $\Delta s(y_i, x)$ and weight vector ω
    Accept transitions y_i where $\alpha_i \leq \Phi$ (hypercone angle)
    Select the best transition y_i^{best} from the candidates
    Adapt Hypercone Angle:
      Compute the rejection rate r_t based on the number of rejected candidate transitions
      Compute the exponential moving average $\overline{r_t}$ of rejection rate
      Update the hypercone angle Φ based on the moving average:

$$\Phi_{t+h} = \text{clip}\left(\Phi_t \exp\left(\eta\left(\overline{r_t} - \tau\right)\right), \Phi_{\min}, \Phi_{\max}\right)$$

  Step 3: Euler Sampling
    Use Euler's method to sample the next state based on the guided velocity field
    Transition to the new sequence
    Update time: $t \to t + h$
end for
Return: Final sequence x_1