AReUReDi: Annealed Rectified Updates for Refining Discrete Flows with Multi-Objective Guidance

Anonymous authors
Paper under double-blind review

Abstract

Designing sequences that satisfy multiple, often conflicting, objectives is a central challenge in therapeutic and biomolecular engineering. Existing generative frameworks largely operate in continuous spaces with single-objective guidance, while discrete approaches lack guarantees for multi-objective Pareto optimality. We introduce **AReUReDi** (**A**nnealed **Re**ctified **U**pdates for **Re**fining **Di**screte Flows), a discrete optimization algorithm with theoretical guarantees of convergence to the Pareto front. Building on Rectified Discrete Flows (ReDi), AReUReDi combines Tchebycheff scalarization, locally balanced proposals, and annealed Metropolis-Hastings updates to bias sampling toward Pareto-optimal states while preserving distributional invariance. Applied to peptide and SMILES sequence design, AReUReDi simultaneously optimizes up to five therapeutic properties (including affinity, solubility, hemolysis, half-life, and non-fouling) and outperforms both evolutionary and diffusion-based baselines. These results establish AReUReDi as a powerful, sequence-based framework for multi-property biomolecule generation.

1 Introduction

The design of biological sequences is inherently multi-objective: therapeutic peptides must balance affinity, immunogenicity, and pharmacokinetics (Naseri & Koffas, 2020; Tominaga et al., 2024); CRISPR guide RNAs trade off on-target activity and off-target risk (Mohr et al., 2016; Schmidt et al., 2025); and synthetic promoters must achieve both strength and tissue specificity (Artemyev et al., 2024). Yet, many computational pipelines still optimize objectives in isolation (Zhou et al., 2019; Nehdi et al., 2020; Nisonoff et al., 2025), often producing adverse trade-offs (e.g., high-affinity peptides that are insoluble or hemolytic) (Bigi et al., 2023; Rinauro et al., 2024). Classical black-box multi-objective optimization (MOO), including evolutionary search and Bayesian optimization, can address such trade-offs (Zitzler & Thiele, 1998; Deb, 2011; Ueno et al., 2016; Frisby & Langmead, 2021) but scales poorly in high-dimensional sequence spaces.

Recent generative methods enable controllable multi-objective sampling (Li et al., 2018; Sousa et al., 2021; Yao et al., 2024). For instance, ParetoFlow (Yuan et al., 2024) produces Pareto-optimal samples via continuous-space flow matching, but extending these ideas to sequences remains challenging: continuous embeddings can obscure token-level structure and complicate property guidance (Beliakov & Lim, 2007; Michael et al., 2024). Discrete flow models offer a more direct alternative by defining probability paths over categorical spaces (Campbell et al., 2024; Gat et al., 2024; Dunn & Koes, 2024), including simplex-based interpolations (Stark et al., 2024; Davis et al., 2024; Tang et al., 2025a) and jump-process transition-rate parameterizations (Campbell et al., 2024; Gat et al., 2024). While promising for single-objective control (Nisonoff et al., 2025; Tang et al., 2025a), Pareto guidance with theoretical guarantees remains largely unexplored.

Here, rectification provides an efficient foundation. In continuous domains, Rectified Flows (Liu et al., 2023) straighten transport trajectories to enable few-step sampling. Recently, **ReDi** (Rectified Discrete Flows) (Yoo et al., 2025) brought this principle to discrete sequences by iteratively refining couplings, reducing factorization error while maintaining distributional fidelity. However, ReDi lacks a mechanism to steer sampling toward the Pareto front, a critical requirement for biomolecular design.

To address this, we introduce **AReUReDi** (**A**nnealed **Re**ctified **U**pdates for **Re**fining **Di**screte Flows), a multi-objective extension of rectified discrete flows. AReUReDi combines (i) *annealed Tchebycheff*

scalarization to progressively emphasize balanced trade-offs (Lin et al., 2024a); (ii) *locally balanced proposals* that fuse the ReDi prior with multi-objective guidance while ensuring reversibility; and (iii) *Metropolis-Hastings updates* that preserve distributional invariance and yield convergence to Pareto-optimal states. Together, these components refine rectified discrete flows into a principled Pareto sampler.

Our key contributions are:

1. We propose AReUReDi, the first multi-objective extension of rectified discrete flows, integrating annealed scalarization, locally balanced proposals, and MCMC updates.
2. We provide theoretical guarantees that AReUReDi preserves distributional invariance and converges to the Pareto front with full coverage.
3. We demonstrate optimization of up to five competing biological properties (affinity, solubility, hemolysis, half-life, and non-fouling) with diverse, biologically plausible designs.
4. We benchmark against classical MOO methods and state-of-the-art discrete diffusion approaches, showing improved trade-off navigation and sample quality.

A detailed discussion of Related Work is provided in Appendix Section A.

2 AReUReDi: Annealed Rectified Updates for Refining Discrete Flows

With an efficient discrete flow-based generation framework in hand, we develop AReUReDi that extends ReDi (Detailed in Appendix Section B) to the multi-objective optimization setting, where the goal is to generate discrete samples that approximate the Pareto front of multiple competing objectives (Yoo et al., 2025). Starting from a pre-trained ReDi model, AReUReDi incorporates annealed guidance, locally balanced proposals, and Metropolis-Hastings updates to progressively bias the sampling process toward Pareto-optimal states while preserving the probabilistic guarantees of the underlying flow (Algorithm 1).

2.1 Problem Setup

Let the discrete search space be $\mathcal{S} = \mathcal{V}^L$, where $\mathcal{V}$ is a finite vocabulary of size K and each state $x = (x_1, \ldots, x_L) \in \mathcal{S}$ is a sequence of tokens. We assume access to a pre-trained ReDi model that provides marginal transition probabilities $p_t^i(\cdot \mid x_t)$ for each position i and time t. In addition, we are given N pre-trained scalar objective functions $s_n : \mathcal{S} \to \mathbb{R}$, where $n = 1, \ldots, N$, and $\tilde{s}_n(x)$ are their normalized counterparts with outputs mapped to $[0, 1]$ to support balanced updates for each objective. The sampling task is to construct a Markov chain whose stationary distribution concentrates on states that approximate the Pareto front of the normalized objectives $\tilde{s}_1, \ldots, \tilde{s}_N$.

2.2 Annealed Multi-Objective Guidance

To direct sampling toward the Pareto front, AReUReDi introduces a scalarized reward

$$S_\omega(x) = \min_{1 \le n \le N} \omega_n \, \tilde{s}_n(x),$$

where the weight vector $\omega = [\omega_1, \ldots, \omega_N]$ lies in the probability simplex Δ^{N-1} and balances the different objectives. This Tchebycheff scalarization promotes solutions that are simultaneously strong across all objectives rather than excelling in only a subset (Miettinen, 1999). The scalarized reward is converted into a guidance weight

$$W_{\eta_t, \omega}(x) = \exp\big(\eta_t S_\omega(x)\big),$$

where the parameter $\eta_t > 0$ controls the strength of the guidance at each iteration t. AReUReDi incorporates an annealing schedule for η_t:

$$\eta_t \;=\; \eta_{\min} + \big(\eta_{\max} - \eta_{\min}\big) \frac{t}{T-1},$$

so that the chain begins with a small value of η_t to encourage wide exploration of the state space and gradually increases η_t to focus sampling on high-quality Pareto candidates. This annealing strategy mirrors simulated annealing but operates directly on the scalarized objectives within the discrete flow framework.

2.3 Locally Balanced Proposals

Given the current state x_t, AReUReDi updates one coordinate $i \in \{1, \dots, L\}$ at a time using a locally balanced proposal that blends the generative prior of ReDi with the multi-objective guidance. First, a candidate set of replacement tokens is drawn from the ReDi marginal $p_t^i(\cdot \mid x_t)$, optionally pruned using top-p to retain only the most promising alternatives for computational efficiency. For each candidate token y, the algorithm computes the ratio

$$r_i(y; x_t) = \frac{W_{\eta_t,\omega}\big(x_t^{(i \leftarrow y)}\big)}{W_{\eta_t,\omega}(x_t)},$$

which measures the change in scalarized reward if x_t^i were replaced by y. The ratio $r_i(y; x_t)$ is then transformed by a balancing function $g : \mathbb{R}_+ \to \mathbb{R}_+$ that satisfies the symmetry condition $g(u) = u\, g(1/u)$. Typical choices include Barker's function $g(u) = \frac{u}{1+u}$ and the square-root function $g(u) = \sqrt{u}$. This symmetry ensures that the resulting Markov chain admits the desired stationary distribution. Using the balanced function, the unnormalized proposal for a candidate token y takes the form

$$\tilde{q}_i(y \mid x_t) = p_t^i(y \mid x_t)\, g\big(r_i(y; x_t)\big),$$

which is then normalized over the candidate set to yield the final proposal distribution $q_i(y \mid x_t)$. This construction allows the proposal to favor states with higher scalarized reward while remaining reversible with respect to the target distribution.

2.4 Metropolis-Hastings Update

A candidate token $y^\star$ is drawn from the final proposal distribution $q_i(\cdot \mid x_t)$ and forms the proposed state $x_{\text{prop}} = x_t^{(i \leftarrow y^\star)}$. The proposal is accepted with the standard Metropolis-Hastings probability (Hastings, 1970)

$$\alpha_i(x_t, x_{\text{prop}}) = \min\left\{1,\ \frac{\pi_{\eta_t,\omega}(x_{\text{prop}})\, q_i(x_t^i \mid x_{\text{prop}})}{\pi_{\eta_t,\omega}(x_t)\, q_i(y^\star \mid x_t)}\right\},$$

where we define $\pi_{\eta_t,\omega}(x) \propto p_1(x)\, W_{\eta_t,\omega}(x) = p_1(x)\, \exp\big(\eta_t S_\omega(x)\big)$. With Barker's balancing function, the acceptance probability simplifies to one, ensuring automatic acceptance of proposals and faster mixing. Other choices, such as the square-root function, trade higher acceptance rates for more conservative moves.

The annealed, locally balanced updates are repeated for T iterations and end with the final sample x_1 whose objective scores are jointly optimized. Building on ReDi's well-calibrated base distribution with low inter-dimensional correlation, AReUReDi biases this base toward Pareto-optimal regions while maintaining support over the full state space. With the standard Metropolis-Hastings acceptance rule, the resulting kernel preserves the reward-tilted target distribution, yielding distributional invariance and asymptotic Pareto-coverage guarantees (Appendix C). We evaluate a monotone-accept variant for finite-budget efficiency in Section 3.

3 Experiments

To the best of our knowledge, no public datasets exist for benchmarking multi-objective optimization algorithms on biological sequences. We therefore developed two benchmarks to evaluate AReUReDi, focusing on the generation of wild-type peptide sequences and chemically-modified peptide SMILES. These tasks are supported by two core components: the generative models described in Appendix D and the objective-scoring models validated in Appendix I. Leveraging these models, we demonstrate AReUReDi's efficacy on a wide range of tasks and examples.

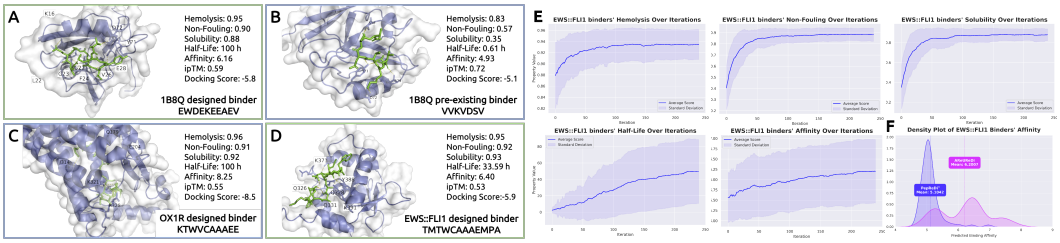

Figure 1: **(A), (B)** Complex structures of PDB 1B8Q with an AReUReDi-designed binder and its pre-existing binder. **(C), (D)** Complex structures of OX1R and EWS::FLI1 with an AReUReDi-designed binder. Five property scores are shown for each binder, along with the ipTM score from AlphaFold3 and docking score from AutoDock VINA. Interacting residues on the target are visualized. **(E)** Plots showing the mean scores for each property across the number of iterations during AReUReDi's design of binders of length 12-aa for EWS::FLI1. **(F)** A density plot illustrating the distribution of predicted property scores for AReUReDi-designed EWS::FLI1 binders of length 12-aa, compared to the peptides generated unconditionally by PepReDi[3].

Although AReUReDi provides theoretical guarantees (Appendix C) for the *unconstrained* MH sampler, these guarantees are asymptotic and may require many sampling steps to manifest. To improve sampling efficiency under a fixed budget, we use a monotone-accept heuristic in all reported experiments, accepting a proposed token update only if it increases the weighted-sum utility. This heuristic makes the resulting Markov chain non-reversible, so the exact invariance and Pareto-coverage guarantees in Appendix C do not strictly apply. Nonetheless, it consistently improves finite-budget optimization performance (Table S6).

3.1 AReUReDi effectively balances objective trade-offs

With a pre-trained PepReDi model, we first validate AReUReDi's ability to balance conflicting objectives. We generate wild-type peptide binders under three-property guidance and perform ablations by removing one or more objectives. For 7LUL (hemolysis, solubility, affinity; Table S7), omitting any guidance collapses the corresponding property, while the remaining guided metrics may improve modestly. For CLK1 (affinity, non-fouling, half-life; Table S8), disabling non-fouling increases half-life beyond 96 h but drives non-fouling near zero, whereas disabling half-life preserves non-fouling but reduces half-life below 2 h. In contrast, enabling all guidance yields the most balanced profiles, indicating that AReUReDi can target specified objectives while navigating trade-offs and steering samples toward the Pareto front.

3.2 AReUReDi generates wild-type peptide binders under five-property guidance

We next benchmark AReUReDi for binder generation guided by five therapeutically relevant properties: hemolysis, non-fouling, solubility, half-life, and binding affinity. We design 100 binders per target for 8 diverse proteins: structured targets with known binders (3IDJ, 5AZ8, 7JVS), structured targets without known binders (AMHR2, OX1R, DUSP12), and intrinsically disordered targets (EWS::FLI1, MYC) (Table S4). Across targets and multiple lengths, AReUReDi achieves strong overall profiles, with high hemolysis scores (0.91-0.94), high non-fouling ($>$0.86) and solubility ($>$0.85), extended half-life (42-64 h), and competitive affinity (5.7-7.3).

For targets with known binders, AReUReDi designs outperform the reference binders across properties while maintaining binding potential (Figure 1A,B, S1), supported by AlphaFold3 ipTM (Abramson et al., 2024) and AutoDock Vina docking (Trott & Olson, 2010). The designs bind similar regions but differ substantially in sequence and structure, indicating broad exploration of sequence space. For targets without known binders, we visualize representative predicted complexes (Figure S2) and report the corresponding property, ipTM, and docking scores, highlighting diverse trade-off patterns (e.g., longer half-life versus improved non-fouling/solubility).

To quantify optimization dynamics, we track the mean and standard deviation of the five scores across 100 length-12 binders for EWS::FLI1 over iterations (Figure 1E). All properties improve steadily, with solubility and non-fouling increasing from $\sim$0.4 to $\sim$0.9. The larger final variance in half-life and affinity reflects their sensitivity under multi-objective trade-offs. Finally, comparing 100 guided

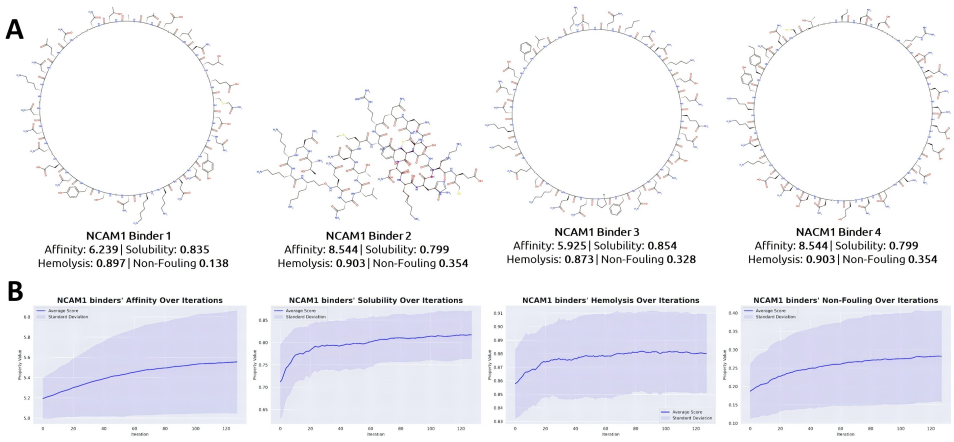

Figure 2: **(A)** Example 2D SMILES structure of AReUReDi-designed peptide binders with four property scores. **(B)** Plots showing the mean scores for each property across the number of iterations during AReUReDi's design of binders of length 200 for NCAM1.

peptides to 100 unconditional samples from PepReDi[3] shows a clear distributional shift toward higher affinity, demonstrating effective multi-property steering.

3.3 AReUReDi generates therapeutic peptide SMILES under four-property guidance

To demonstrate broad applicability beyond wild-type sequences, we apply AReUReDi to the rectified SMILESReDi model to design chemically modified peptide-binder SMILES for five therapeutic targets: GLP1R, TfR, NCAM1, GLAST, and AMHR2. For each target, generation is jointly guided by four objectives (binding affinity, hemolysis, solubility, and non-fouling) to promote both potency and favorable physicochemical profiles. While PepTune can also generate peptide-binder SMILES, it does not report average property scores for its designs, preventing direct quantitative comparison (Tang et al., 2025b).

We visualize the top-affinity binder for each target (Figure 2A, S3A,C, S4A,C); all achieve strong scores across the four objectives. To assess optimization dynamics, we track the mean and standard deviation of each property over 100 generated binders per iteration (Figure 2B, S3B,D, S4B,D). Across targets, affinity and non-fouling improve steadily, while hemolysis and solubility fluctuate, reflecting trade-off balancing under multi-objective guidance. Finally, AReUReDi produces valid SMILES with higher diversity and lower SNN than PepTune (Table S3), indicating greater novelty and structural variability. Overall, these results show that AReUReDi reliably enables *de novo* multi-property design of chemically modified peptide binders across diverse targets.

A detailed benchmarking against MOO and diffusion baselines is provided in Appendix Section E, and a detailed discussion of ablation studies is provided in Appendix Section F.

4 Discussion

We presented **AReUReDi**, a multi-objective framework that extends rectified discrete flows for biomolecular sequence design under competing property constraints. By combining annealed Tchebycheff scalarization, locally balanced proposals, and Metropolis-Hastings updates, AReUReDi offers theoretical guarantees of Pareto-front convergence with full-coverage sampling, and achieves strong *in silico* performance across both amino-acid sequences and chemically modified peptide SMILES. Future work includes extending AReUReDi to other modalities (e.g., DNA/RNA, antibodies, and combinatorial libraries) and improving efficiency without sacrificing theoretical guarantees, including uncertainty-aware and feedback-driven guidance. Overall, AReUReDi provides a principled foundation for multi-property-guided biomolecular design aimed at producing therapeutics that are both potent and explicitly optimized for clinical developability.

REPRODUCIBILITY STATEMENT

We ensure reproducibility through detailed theoretical, algorithmic, and experimental descriptions of AReUReDi. The complete procedure is formally described in the main text with proofs of convergence guarantees, including the rectified discrete flow foundation, annealed Tchebycheff scalarization, locally balanced proposals, and Metropolis-Hastings updates. Architectures, training details, and datasets for all base generators (PepReDi, SMILESReDi, and PepDFM) are reported with quantitative metrics in the Results and Appendix. Hyperparameter settings, annealing schedules, and sensitivity analyses are provided to facilitate replication, and ablation studies are included to assess the impact of key design choices. Benchmark comparisons against classical multi-objective optimization baselines and diffusion-based methods are tabulated for reference. All datasets used in this work (PepNN, BioLip2, PPIRef, peptide property datasets, and peptide SMILES collections) are publicly available. We will release code, pretrained checkpoints, and sampling scripts for AReUReDi to enable full reproducibility.

ETHICS STATEMENT

This work develops a general generative modeling framework for multi-objective sequence design, with demonstrations on peptide and peptide-SMILES generation. All datasets are publicly available and non-sensitive, consisting of peptide property measurements, protein-peptide interaction sets, and peptide SMILES representations. No human subjects, patient data, or animal experiments were involved. Potential risks include the misuse of generative models for harmful molecule design or the uncontrolled release of potent sequences. To mitigate these risks, we will release code and pretrained models strictly under a research-only license and provide documentation that emphasizes safe and responsible use. The anticipated societal benefits, such as improving therapeutic peptide design, enhancing drug safety profiles, and enabling efficient exploration of biological sequence space, substantially outweigh these potential risks. We encourage future users of AReUReDi to adopt similar safeguards when applying the method to other molecular domains.

REFERENCES

Osama Abdin, Satra Nim, Han Wen, and Philip M Kim. Pepnn: a deep attention model for the identification of peptide binding sites. *Communications biology*, 5(1):503, 2022.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Alaleh Ahmadianshalchi, Syrine Belakaria, and Janardhan Rao Doppa. Pareto front-diverse batch multi-objective bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10784–10794, 2024.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019.

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara E Engelhardt. Preference-guided diffusion for multi-objective offline optimization. *arXiv preprint arXiv:2503.17299*, 2025.

Valentin Artemyev, Anna Gubaeva, Anastasiia Iu Paremskaia, Amina A Dzhioeva, Andrei Deviatkin, Sofya G Feoktistova, Olga Mityaeva, and Pavel Yu Volchkov. Synthetic promoters in gene therapy: Design approaches, features and applications. *Cells*, 13(23):1963, 2024.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Output space entropy search framework for multi-objective bayesian optimization. *Journal of artificial intelligence research*, 72:667–715, 2021.

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

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

Alessandra Bigi, Eva Lombardo, Roberta Cascella, and Cristina Cecchi. The toxicity of protein aggregates: new insights into the mechanisms, 2023.

Jenny Bostrom, Chingwei V Lee, Lauric Haber, and Germaine Fuh. Improving antibody binding affinity and specificity for therapeutic development. In *Therapeutic Antibodies: Methods and Protocols*, pp. 353–376. Springer, 2008.

Anton Bushuiev, Roman Bushuiev, Petr Kouba, Anatolii Filkin, Marketa Gabrielova, Michal Gabriel, Jiri Sedlar, Tomas Pluskal, Jiri Damborsky, Stanislav Mazurenko, et al. Learning to design protein-protein interactions with enhanced generalization. *arXiv preprint arXiv:2310.18515*, 2023.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=kQwSbv0BR4`.

Shengfu Chen, Zhiqiang Cao, and Shaoyi Jiang. Ultra-low fouling peptide surfaces derived from natural amino acids. *Biomaterials*, 30(29):5892–5896, 2009.

CA Coello Coello and Maximino Salazar Lechuga. 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.

Oscar Davis, Samuel Kessler, Mircea Petrache, Ismail Ceylan, Michael Bronstein, and Joey Bose. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

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

Kalyanmoy Deb and Himanshu Jain. 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.

Ian Dunn and David Ryan Koes. Exploring discrete flow matching for 3d de novo molecule generation. *ArXiv*, pp. arXiv–2411, 2024.

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

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Rapport technique, Leiden University*, 34: 7–3, 2008.

Keld Fosgerau and Torsten Hoffmann. Peptide therapeutics: current status and future directions. *Drug discovery today*, 20(1):122–128, 2015.

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

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

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

Xu Han, Caihua Shan, Yifei Shen, Can Xu, Han Yang, Xiang Li, and Dongsheng Li. Training-free multi-objective diffusion model for 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2023.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International conference on machine learning*, pp. 14631–14653. PMLR, 2023.

Shipra Jain, Srijanee Gupta, Sumeet Patiyal, and Gajendra PS Raghava. Thpdb2: compilation of fda approved therapeutic peptides and proteins. *Drug Discovery Today*, pp. 104047, 2024.

Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, 10(1): 50–66, 2006.

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

Xi Lin, Yilu Liu, Xiaoyuan Zhang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Few for many: Tchebycheff set scalarization for many-objective optimization. *arXiv preprint arXiv:2405.19650*, 2024a.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. *arXiv preprint arXiv:2402.19078*, 2024b.

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.

Meitong Liu, Xiaoyuan Zhang, Chulin Xie, Kate Donahue, and Han Zhao. Online mirror descent for tchebycheff scalarization in multi-objective optimization. *arXiv preprint arXiv:2410.21764*, 2024.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Deepika Mathur, Satya Prakash, Priya Anand, Harpreet Kaur, Piyush Agrawal, Ayesha Mehta, Rajesh Kumar, Sandeep Singh, and Gajendra PS Raghava. Peplife: a repository of the half-life of peptides. *Scientific reports*, 6(1):36617, 2016.

Richard Michael, Simon Bartels, Miguel González-Duque, Yevgen Zainchkovskyy, Jes Frellsen, Søren Hauberg, and Wouter Boomsma. A continuous relaxation for discrete bayesian optimization. *arXiv preprint arXiv:2404.17452*, 2024.

K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer, Boston, USA, 1999.

Stephanie E Mohr, Yanhui Hu, Benjamin Ewen-Campen, Benjamin E Housden, Raghuvir Viswanatha, and Norbert Perrimon. Crispr guide rna design for research applications. *The FEBS journal*, 283 (17):3232–3238, 2016.

Gita Naseri and Mattheos AG Koffas. Application of combinatorial optimization strategies in synthetic biology. *Nature communications*, 11(1):2446, 2020.

Atef Nehdi, Nosaibah Samman, Vanessa Aguilar-Sánchez, Azer Farah, Emre Yurdusev, Mohamed Boudjelal, and Jonathan Perreault. Novel strategies to optimize the amplification of single-stranded dna. *Frontiers in Bioengineering and Biotechnology*, 8:401, 2020.

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

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pp. 766–776. PMLR, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, November 2011. ISSN 1532-4435.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

M Pirtskhalava, B Vishnepolsky, and M Grigolava. Transmembrane and antimicrobial peptides. hydrophobicity, amphiphilicity and propensity to aggregation. *arXiv preprint arXiv:1307.6160*, 2013.

Dillon J Rinauro, Fabrizio Chiti, Michele Vendruscolo, and Ryan Limbocker. Misfolded protein oligomers: Mechanisms of formation, cytotoxic effects, and pharmacological approaches against protein misfolding diseases. *Molecular Neurodegeneration*, 19(1):20, 2024.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 2024.

Henri Schmidt, Minsi Zhang, Dimitar Chakarov, Vineet Bansal, Haralambos Mourelatos, Francisco J Sánchez-Rivera, Scott W Lowe, Andrea Ventura, Christina S Leslie, and Yuri Pritykin. Genome-wide crispr guide rna design and specificity analysis with guidescan2. *Genome biology*, 26(1): 1–25, 2025.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Tiago Sousa, João Correia, Vitor Pereira, and Miguel Rocha. 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.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International conference on machine learning*, pp. 20459–20478. PMLR, 2022.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ruiqing Sun, Dawei Feng, Sen Yang, Yijie Wang, and Huaimin Wang. Evolutionary training-free guidance in diffusion model for 3d multi-objective molecular generation. *arXiv preprint arXiv:2505.11037*, 2025.

R Swanson. Long live peptides—evolution of peptide half-life extension technologies and emerging hybrid approaches. *Drug Discovery World*, 15:57–61, 2014.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Gumbel-softmax flow matching with straight-through guidance for controllable biological sequence generation. *arXiv preprint arXiv:2503.17361*, 2025a.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2025b.

Masahiro Tominaga, Yoko Shima, Kenta Nozaki, Yoichiro Ito, Masataka Someda, Yuji Shoya, Noritaka Hashii, Chihiro Obata, Miho Matsumoto-Kitano, Kohei Suematsu, et al. Designing strong inducible synthetic promoters in yeasts. *Nature Communications*, 15(1):10653, 2024.

Oleg Trott and Arthur J Olson. 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.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444, 2023.

Ben Tu, Nikolas Kantas, Robert M Lee, and Behrang Shafei. Multi-objective optimisation via the r2 utilities. *arXiv preprint arXiv:2305.11774*, 2023.

Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Teruyasu Mizoguchi, and Koji Tsuda. Combo: An efficient bayesian optimization library for materials science. *Materials discovery*, 4:18–21, 2016.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024.

Jing Yang, S Mark Roe, Matthew J Cliff, Mark A Williams, John E Ladbury, Patricia T W Cohen, and David Barford. Molecular basis for tpr domain-mediated regulation of protein phosphatase 5. *The EMBO Journal*, 24(1):1–10, December 2004. ISSN 1460-2075. doi: 10.1038/sj.emboj.7600496. URL http://dx.doi.org/10.1038/sj.emboj.7600496.

Yinghua Yao, Yuangang Pan, Jing Li, Ivor Tsang, and Xin Yao. Proud: Pareto-guided diffusion model for multi-objective generation. *Machine Learning*, 113(9):6511–6538, 2024.

Jaehoon Yoo, Wonjung Kim, and Seunghoon Hong. Redi: Rectified discrete flow. *arXiv preprint arXiv:2507.15897*, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Ye Yuan, Can Chen, Christopher Pal, and Xue Liu. Paretoflow: Guided flows in multi-objective optimization. *arXiv preprint arXiv:2412.03718*, 2024.

Chengxin Zhang, Xi Zhang, Peter L Freddolino, and Yang Zhang. Biolip2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, 52(D1): D404–D412, 2024.

Mingjie Zhang, Hidehito Tochio, Qiang Zhang, Pravat Mandal, and Ming Li. Solution structure of the extended neuronal nitric oxide synthase pdz domain complexed with an associated peptide. *Nature Structural Biology*, 6(5):417–421, May 1999. ISSN 1072-8368. doi: 10.1038/8216. URL http://dx.doi.org/10.1038/8216.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Ruochi Zhang, Haoran Wu, Yuting Xiu, Kewei Li, Ningning Chen, Yu Wang, Yan Wang, Xin Gao, and Fengfeng Zhou. 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.

Ruimin Zhou, Zhaoyan Jiang, Chen Yang, Jianwei Yu, Jirui Feng, Muhammad Abdullah Adil, Dan Deng, Wenjun Zou, Jianqi Zhang, Kun Lu, et al. All-small-molecule organic solar cells with over 14% efficiency by optimizing hierarchical morphologies. *Nature communications*, 10(1):5393, 2019.

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

Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103, 2001.

Appendix

A Related Works

Online Multi-Objective Optimization. Recent work in multi-objective guided generation has focused on online or sequential decision-making, where solutions are refined with new data (Gruver et al., 2023; Jain et al., 2023; Stanton et al., 2022; Ahmadianshalchi et al., 2024). A common approach is Bayesian optimization (BO), which builds a surrogate model and proposes evaluations via acquisition functions (Yu et al., 2020; Shahriari et al., 2015). Multi-objective BO often uses advanced criteria such as EHVI (Emmerich & Klinkenberg, 2008), information gain (Belakaria et al., 2021), or scalarization (Knowles, 2006; Zhang & Li, 2007; Paria et al., 2020). While AReUReDi also employs Tchebycheff scalarization, it operates in an offline setting, where each sequence requires costly evaluation. This contrasts with the sequential, feedback-driven nature of online methods, making direct comparison inappropriate.

Tchebycheff Scalarization. Tchebycheff scalarization can identify any Pareto-optimal point and is widely used in multi-objective optimization (Miettinen, 1999). Recent variants include smooth scalarization for gradient-based algorithms (Lin et al., 2024b) and OMD-TCH for online learning (Liu et al., 2024). AReUReDi is, to our knowledge, the first to apply Tchebycheff scalarization for offline generative design of discrete therapeutic sequences. Future work may extend to many-objective problems or alternative utility functions (Lin et al., 2024a; Tu et al., 2023).

Diffusion and Flow Matching. Generative approaches such as ParetoFlow and PGD-MOO adapt flow matching or diffusion models for multi-objective optimization (Yuan et al., 2024; Annadani et al., 2025). These operate in continuous or latent spaces, whereas AReUReDi is designed for discrete token spaces inherent to biological sequences. This domain mismatch precludes direct benchmarking.

Biomolecule Generation. Offline multi-objective frameworks such as EGD and MUDM have optimized molecules with multiple properties (Sun et al., 2025; Han et al., 2023), but these emphasize 3D structural representations. By contrast, AReUReDi is sequence-only, operating directly over amino acids or SMILES, which makes structural methods unsuitable as direct comparators.

B Preliminaries

B.1 Discrete Flow Matching

Let $\mathcal{S} = V^L$ denote the discrete state space, where V is a vocabulary of size K and each $x = (x_1, \ldots, x_L) \in \mathcal{S}$ is a sequence of tokens. A *discrete flow matching (DFM)* model (Campbell et al., 2024; Gat et al., 2024; Dunn & Koes, 2024) defines a probability path $\{p_t\}_{t\in[0,1]}$ interpolating between a simple source distribution p_0 and a target distribution p_1 by means of a coupling $\pi(x_0, x_1)$ and conditional bridge distributions $p_t(x_t \mid x_0, x_1)$. The model is trained to approximate conditional transitions $p_{s|t}(x_s \mid x_t)$ for $0 \le t < s \le 1$.

Since the joint distribution over L coordinates is intractable, DFMs employ a factorization

$$p_{s|t}(x_s \mid x_t) \;\approx\; \prod_{i=1}^{L} p_{s|t}\big(x_s^i \mid x_t\big)\,,$$

which introduces a discrepancy measured by the conditional total correlation

$$\mathrm{TC}_{s|t} \;=\; \mathrm{KL}\left(p_{s|t}(x_s \mid x_t) \;\middle\|\; \prod_{i=1}^{L} p_{s|t}(x_s^i \mid x_t) \right).$$

This quantity captures the inter-dimensional dependencies neglected under factorization, and grows with larger step sizes (Stark et al., 2024; Davis et al., 2024; Tang et al., 2025a). As a result, DFMs are accurate in the many-step regime but degrade under few-step or one-step generation.

B.2 Rectified Discrete Flow

To mitigate factorization error, **Rectified Discrete Flow (ReDi)** (Yoo et al., 2025) introduces an iterative rectification of the coupling π. Starting from an initial coupling $\pi^{(0)}(x_0, x_1)$, a DFM is trained under $\pi^{(k)}$ to produce new source-target pairs, defining an empirical joint distribution $\hat{\pi}^{(k)}$. The coupling is then updated via

$$\pi^{(k+1)}(x_0, x_1) \;\propto\; \pi^{(k)}(x_0, x_1)\, \frac{p_{\theta^{(k)}}(x_1 \mid x_0)}{p_{\theta^{(k)}}(x_1)},$$

where $p_{\theta^{(k)}}(x_1 \mid x_0)$ is the conditional distribution learned at iteration k. This yields a sequence of couplings $\{\pi^{(k)}\}_{k\geq 0}$ with provably decreasing conditional TC,

$$\mathrm{TC}_{s|t}(\pi^{(k+1)}) \;\leq\; \mathrm{TC}_{s|t}(\pi^{(k)}).$$

By progressively reducing factorization error, ReDi produces a well-calibrated base distribution p_1 with low inter-dimensional correlation. This base distribution provides reliable marginal transition probabilities $p_t^i(\cdot \mid x_t)$ for each coordinate i at time t, which serve as the generative prior in the AReUReDi framework. Rectification follows the same principle as *Rectified Flow* in continuous domains (Liu et al., 2023), where iterative refinement straightens ODE paths and decreases transport costs.

C Theoretical Guarantees

In this section, we establish that AReUReDi converges to Pareto-optimal solutions while preserving coverage of the entire Pareto front. We assume throughout that the state space $\mathcal{S}$ is finite, all objective functions s_n are bounded, and their normalized versions $\tilde{s}_n$ map to $[0, 1]$.

C.1 Preliminary Definitions

Definition (Pareto Optimality). A state $x^* \in \mathcal{S}$ is *Pareto optimal* if there exists no $y \in \mathcal{S}$ such that $\tilde{s}_n(y) \geq \tilde{s}_n(x^*)$ for all $n \in \{1, \ldots, N\}$ with strict inequality for at least one n.

Definition (Pareto Front). The Pareto front is $\mathcal{P} = \{x \in \mathcal{S} : x \text{ is Pareto optimal}\}$.

Definition (Interior Weight Vector). A weight vector $\omega \in \Delta^{N-1}$ is *interior* if $\omega_n > 0$ for all n.

C.2 Main Theoretical Results

Theorem (Invariance). The Markov kernel defined by the Locally Balanced Proposal (LBP) and Metropolis-Hastings update leaves the distribution

$$\pi_{\eta,\omega}(x) \propto p_1(x) \exp\big(\eta S_\omega(x)\big)$$

invariant for every guidance strength $\eta > 0$ and weight vector $\omega \in \Delta^{N-1}$.

Proof. We prove this in two steps: first showing that single-coordinate updates preserve detailed balance, then that random-scan mixtures preserve invariance.

Step 1: Single-coordinate detailed balance. Let x and x' differ only at coordinate i, where $x'_i = y$ for some token y. The proposal probability is

$$q_i(y \mid x) = \frac{p_t^i(y \mid x_t) g(r_i(y; x_t))}{\sum_{z \in \text{candidates}} p_t^i(z \mid x_t) g(r_i(z; x_t))},$$

where $r_i(y; x_t) = \frac{W_{\eta_t,\omega}(x_t^{(i \leftarrow y)})}{W_{\eta_t,\omega}(x_t)}$ and g satisfies $g(u) = u \cdot g(1/u)$.

The acceptance probability is

$$\alpha_i(x, x') = \min\left\{1, \frac{\pi_{\eta,\omega}(x') q_i(x_i \mid x')}{\pi_{\eta,\omega}(x) q_i(y \mid x)}\right\}.$$

By the symmetry property of g and the construction of the proposal, we have

$$\frac{q_i(y \mid x)}{q_i(x_i \mid x')} = \frac{W_{\eta,\omega}(x')}{W_{\eta,\omega}(x)}.$$

Since $\pi_{\eta,\omega}(x) = Z^{-1} p_1(x) W_{\eta,\omega}(x)$, it follows that

$$\frac{\pi_{\eta,\omega}(x') q_i(x_i \mid x')}{\pi_{\eta,\omega}(x) q_i(y \mid x)} = 1.$$

Therefore, $\alpha_i(x, x') = 1$ and detailed balance is satisfied.

Step 2: Random-scan mixture. The overall kernel is $K(x, x') = \frac{1}{L} \sum_{i=1}^{L} K_i(x, x')$, where K_i is the kernel for updating coordinate i. Since each K_i satisfies detailed balance with respect to $\pi_{\eta,\omega}$, their convex combination also satisfies detailed balance and hence preserves invariance. □

Theorem (Convergence to Pareto Front). Fix any $\omega \in \operatorname{int} \Delta^{N-1}$ with strictly positive entries and let $S_\omega(x) = \min_n \omega_n \tilde{s}_n(x)$. If $\eta \to \infty$, samples drawn from $\pi_{\eta,\omega}(x) \propto p_1(x) \exp(\eta S_\omega(x))$ concentrate on the set

$$\mathcal{F}_\omega = \arg\max_x S_\omega(x),$$

and every element of $\mathcal{F}_\omega$ is Pareto optimal.

Proof. **Step 1: Maximizers of S_ω are Pareto optimal.** Suppose $x^* \in \mathcal{F}_\omega$ but x^* is not Pareto optimal. Then there exists $y \in \mathcal{S}$ with

$$\tilde{s}_n(y) \geq \tilde{s}_n(x^*) \; \forall n, \quad \text{and} \quad \tilde{s}_m(y) > \tilde{s}_m(x^*) \text{ for some } m.$$

Since $\omega_n > 0$ for all n, multiplying preserves inequalities. If m is the bottleneck coordinate of x^*, then $S_\omega(y) > S_\omega(x^*)$, contradiction. Otherwise, equality requires special weight alignments (measure zero). Thus maximizers are Pareto optimal almost surely.

Step 2: Concentration as $\eta \to \infty$. Let $S^*_\omega = \max_x S_\omega(x)$ and $\Delta_\omega = S^*_\omega - \max_{x \notin \mathcal{F}_\omega} S_\omega(x) > 0$. Then for $x \notin \mathcal{F}_\omega$,

$$\pi_{\eta,\omega}(x) \leq e^{-\eta \Delta_\omega} \cdot \frac{p_1(x)}{\sum_{z \in \mathcal{F}_\omega} p_1(z)}.$$

Summing gives $\pi_{\eta,\omega}(\mathcal{S} \setminus \mathcal{F}_\omega) \to 0$ as $\eta \to \infty$. Hence the mass concentrates on $\mathcal{F}_\omega$. □

Theorem (Pareto Point Representability). For every Pareto-optimal state $x^\dagger \in \mathcal{P}$ there exists $\omega \in \Delta^{N-1}$ such that $x^\dagger \in \arg\max_x S_\omega(x)$. Moreover, if $\tilde{s}_n(x^\dagger) > 0$ for all n, then $x^\dagger$ can be made the unique maximizer.

Proof. If $\tilde{s}_n(x^\dagger) > 0$, define

$$\omega_n = \frac{1/\tilde{s}_n(x^\dagger)}{\sum_{k=1}^{N} 1/\tilde{s}_k(x^\dagger)}.$$

Then $S_\omega(x^\dagger) = \frac{1}{\sum_k 1/\tilde{s}_k(x^\dagger)}$, and for any $y \neq x^\dagger$, some m satisfies $\tilde{s}_m(y) < \tilde{s}_m(x^\dagger)$, implying $S_\omega(y) < S_\omega(x^\dagger)$. If some $\tilde{s}_n(x^\dagger) = 0$, perturb objectives by $\varepsilon > 0$ and take the limit. □

Theorem (Coverage Guarantee). Let μ be any probability distribution with full support on $\operatorname{int} \Delta^{N-1}$. If $\omega \sim \mu$ and $\eta \to \infty$, then the induced sampler visits every Pareto-optimal state with positive probability.

Proof. By representability, each Pareto point $x^\dagger$ maximizes S_ω for some interior $\omega^\dagger$. By continuity, there exists a neighborhood $U_{x^\dagger}$ where $x^\dagger$ remains optimal. Since $\mu(U_{x^\dagger}) > 0$, randomizing ω ensures $x^\dagger$ is visited with positive probability in the high-η limit. □

Remark. The guarantees hold for any finite $\mathcal{S}$ and bounded objectives. In practice, convergence depends on the chain mixing rate, the annealing schedule for η, and the choice of balancing function g.

D PepReDi and SMILESReDi Generate Diverse and Biologically Plausible Sequences

To enable the efficient generation of peptide binders, we developed an unconditional peptide generator, **PepReDi**, based on the ReDi framework. The model backbone of PepReDi is a Diffusion Transformer (DiT) architecture (Peebles & Xie, 2022). We trained PepDFM on a custom dataset comprising approximately 15,000 peptides from the PepNN and BioLip2 datasets, as well as sequences from the PPIRef dataset, with lengths ranging from 6 to 49 amino acids (Abdin et al., 2022; Zhang et al., 2024; Bushuiev et al., 2023). Using this trained model, we generated new data couplings containing 10,000 sequences for each peptide length and used them to fine-tune PepReDi in an iterative rectification procedure. This rectification was performed three times and yielded substantial improvements in training loss, validation negative log-likelihood (NLL), perplexity (PPL), and conditional TC (Table S2). Notably, the conditional TC rises after the first rectification, likely due to the distributional shift from the large, model-generated coupling, whose absolute TC can be higher even though ReDi guarantees a monotonic decrease within each coupling. The low validation NLL and PPL metrics showcase PepReDi's reliability to generate biologically plausible wild-type peptide sequences.

SMILESReDi adopts the same backbone structure as PepReDi, enhanced with Rotary Positional Embeddings (RoPE), which effectively captures the relative inter-token interactions in peptide SMILES (Su et al., 2024). SMILESReDi also incorporates a time-dependent noising schedule to improve its capability to generate valid peptide SMILES sequences (G.2). We applied the same training data as PepMDLM, a state-of-the-art diffusion model that generates valid peptide SMILES sequences (Tang et al., 2025b). After only two training epochs, SMILESReDi converged to a validation NLL of 0.722 and achieved a sampling validity of 76.3% using just 16 generation steps. One hundred SMILES sequences were then generated by the trained SMILESReDi for each length from 4 to 1035, forming a large and diverse new data coupling. Following a single round of rectification, the validation NLL further decreased to 0.608, and the sampling validity rose dramatically to 98.6% with 16 steps and 100% with 32 steps S3. While its similarity-to-nearest-neighbor (SNN) score and diversity are comparable to those of PepMDLM (details on metrics are provided in Appendix G.2), SMILESReDi substantially outperforms PepMDLM in validity, highlighting its superior capability of generating diverse chemically-modified peptide SMILES sequences.

E AReUReDi Benchmarking Against MOO and Diffusion Baselines

We benchmarked AReUReDi against four established multi-objective optimization (MOO) baselines (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 (Zhang et al., 1999), and PPP5, a larger protein without characterized binders (Yang et al., 2004) (Table S5). Each method generated 100 candidate binders optimized for five properties: hemolysis, non-fouling, solubility, half-life, and binding affinity. While AReUReDi required longer runtimes than evolutionary baselines, it consistently produced the best trade-offs. For both targets, it designed targets with top hemolysis scores, increased non-fouling and solubility by 30-50%, maintained competitive binding affinity, and even extended the half-life by a factor of 3-13 relative to the next-best method. These results underscore AReUReDi's effectiveness in navigating high-dimensional property landscapes to yield peptide binders with balanced, optimized profiles.

We also compared against PepTune (Tang et al., 2025b), a recent masked discrete diffusion model for peptide design that couples generation with Monte Carlo Tree Search for MOO. PepTune's backbone was adapted to the existing DPLM model (Wang et al., 2024) for wild-type peptide sequence generation. Despite longer runtimes, AReUReDi substantially outperformed PepTune across all objectives, yielding nearly threefold improvements in non-fouling and solubility and a 22-fold increase in half-life. Together, these comparisons demonstrate that AReUReDi surpasses not only traditional MOO algorithms but also the current state-of-the-art diffusion-based approach for multi-objective-guided wild-type peptide binder design.

Since AReUReDi requires more computation than PepTune to design the same number of binders, we compare both methods under a matched wall-clock budget (Table S11). Specifically, the time PepTune needs to generate 100 binders approximately matches the time AReUReDi needs to generate four 8-mer binders for 1B8Q and three 16-mer binders for PPP5. For both tasks, the top-2 AReUReDi binders

achieve substantially higher non-fouling, solubility, and half-life, while maintaining comparable hemolysis and affinity. This comparison shows that AReUReDi produces better multi-objective trade-offs, even when PepTune is allowed a much larger candidate pool under the same time budget.

F Ablation Studies

Effect of Rectification. To determine whether rectification improves over standard discrete flow matching, we compare AReUReDi using three generators: the base PepReDi model (no rectification), PepReDi after three rounds of rectification, and PepDFM, a standard discrete flow matching model trained on the same data following Gat et al. (2024) (Appendix G.3). We design wild-type binders for two distinct protein targets, 5AZ8 and AMHR2 (Table S9). For AMHR2, the rectified model achieves the best scores across all five properties, and improves predicted half-life by nearly 13 hours over the next-best method. For 5AZ8, rectification yields a substantially higher half-life while maintaining comparable performance on the remaining metrics. These results suggest that by lowering conditional total correlation and improving the probability path quality, rectification enables stronger Pareto trade-offs on the most demanding objectives.

Annealed vs. Fixed Guidance Strength. We next ablate the guidance strength schedule by comparing the default annealed η_t against fixed guidance strengths set to $\eta_{\min}$, $\eta_{\max}$, and their midpoint (Table S10). We evaluate on two settings: a structured protein with known binders (PDB 1DDV) and an intrinsically disordered protein without known binders (P53). Across both targets, no fixed-η setting matches the annealed schedule. For 1DDV, annealing yields markedly higher half-life and the best solubility, while maintaining hemolysis, non-fouling, and affinity that meet or exceed all fixed-η settings. The same trend holds for P53, where annealing consistently delivers the strongest performance across objectives. Overall, gradually increasing guidance strength improves Pareto trade-offs, boosting challenging properties such as half-life without sacrificing other therapeutic metrics.

Computational Budget. To characterize the quality-compute trade-off, we vary the number of sampling steps on a wild-type binder design task (MYC, 12-mers) and a chemically-modified binder design task (NCAM1, length 200) (Table S12). Increasing the step budget improves the optimized properties in both settings, while runtime scales approximately linearly. For the more expensive SMILES task, improvements from 128 to 256 steps are marginal relative to the added compute, motivating our default choice of 128-256 steps for wild-type binders and 128 steps for chemically-modified binders in the main experiments (Appendix J).

Sensitivity to Weight Vectors. Finally, we verify that changing the Tchebycheff weight vector ω steers AReUReDi to different regions of the Pareto front by varying ω on two three-objective tasks: wild-type peptide binder design for CLK1 and chemically-modified binder design for GFAP (Tables S13, S14). Balanced weights yield balanced improvements, while emphasizing a single objective systematically shifts designs toward that objective with corresponding trade-offs in the others. This confirms that ω provides controllable navigation of Pareto trade-offs rather than merely re-sampling a fixed compromise.

Role of the ReDi Prior. We ablate the ReDi prior by replacing the learned prior $p_1(x)$ with an uniform prior while keeping the rest of AReUReDi unchanged, across wild-type binder (PPP5, 1B8Q) and chemically-modified binder (TfR, GLP1) design tasks (Table S15, S16). Using the learned prior consistently improves multi-objective outcomes, indicating that the flow prior is not redundant in the reward-tilted sampler but instead anchors generation in realistic, high-quality regions of sequence space.

G Base Model Details

G.1 PepReDi

Model Architecture. The backbone of PepReDi is built on a Diffusion Transformer (DiT) framework implemented within a Masked Diffusion Language Model (MDLM) paradigm (Peebles & Xie, 2022;

Sahoo et al., 2024). Input amino acid sequences are transformed to discrete tokens using the ESM-2-650M tokenizer (Lin et al., 2023). Tokenized amino acid sequences and time-steps are converted to continuous embedding vectors using two separate layers, which are then fused and processed by stacked DiT transformer blocks equipped with multi-head self-attention to capture long-range dependencies in the amino-acid sequence. Residual connections and layer normalization stabilize the training dynamics, and a final projection layer outputs token logits for each position.

Dataset Curation. The dataset for PepReDi 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. Training was conducted on a single node equipped with one NVIDIA GPU and 128 GB of GPU memory using the SLURM workload manager. The model was trained for 100 epochs using the Adam optimizer and a learning rate of 1e-4 with weight decay of 1e-5. A learning rate scheduler with 10 warm-up epochs and cosine decay was used, with initial and minimum learning rates both 1e-5. The network architecture included a model dimension of 512, 6 transformer layers, and 8 attention heads, with a vocabulary size of 24 and a maximum sequence length of 100 tokens. Conditional total correlation estimation was performed using 20 batches and 50 samples per batch to monitor rectification quality during training. The model checkpoint with the lowest total correlation was saved. For training rectified models, the same hyperparameter setting was applied, except for the loaded pre-trained model checkpoint and the weight decay being increased to 2e-5.

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.

Rectification. The trained model applied 16 sampling steps to generate 10k sequences for each peptide length, ranging from 6 to 49, with a temperature hyperparameter set to 1. After generation, dynamic batching was used to optimize GPU memory utilization for future rectified training.

G.2 SMILESREDI

Model Architecture. SMILESReDi follows the ReDi paradigm and uses a Diffusion Transformer (DiT) backbone embedded in a Masked Diffusion Language Model (MDLM) design to generate molecular SMILES sequences (Peebles & Xie, 2022; Sahoo et al., 2024). Input SMILES sequences are transformed to discrete tokens using the PeptideCLM -23M tokenizer. Tokenized amino acid sequences and time-steps are converted to continuous embedding vectors using two separate layers. Both embeddings are then fused and processed by stacked DiT transformer blocks that incorporate Rotary Positional Embeddings (RoPE) and multi-head attention modules to capture long-range structural dependencies while preserving positional information (Su et al., 2024). A final layer normalization and linear projection outputs token logits for each position.

Time-dependent bond-aware noising schedule. Peptide SMILES share a conserved backbone of alternating carbonyl and amide groups connected by chemically constrained peptide bonds, while their side chains remain highly diverse. Standard discrete flow matching can corrupt these critical bond tokens too early, hindering the flow from recovering the backbone along the probability path. Inspired by previous work in bond-dependent masking, we devised a time-dependent bond-aware noising schedule that preserves backbone tokens longer than side-chain tokens, allowing the model to reconstruct the invariant scaffold before generating variable side chains. Specifically, for each position j with a bond indicator $b_j \in \{0, 1\}$, the time-t marginal of the probability path is

$$p_t(x_t^{(j)} \mid x_0^{(j)}, x_1^{(j)}) = \left[\, b_j t^\gamma + (1 - b_j)t \,\right] \delta_{x_1^{(j)}} + \left[\, 1 - b_j t^\gamma - (1 - b_j)t \,\right] \delta_{x_0^{(j)}}, \quad t \in [0, 1],\ \gamma > 1,$$

so each token is equal to $x_1^{(j)}$ with the indicated mixture coefficient and to $x_0^{(j)}$ otherwise, ensuring that backbone tokens ($b_j = 1$) transition more slowly than non-bond tokens along the DFM probability path.

Training Strategy. The training is conducted on a 4*A6000 NVIDIA RTX 6000 Ada GPU system with 48 GB of VRAM for 5 epochs. The model checkpoint with the lowest evaluation loss was saved. The Adam optimizer was employed with a learning rate of 1e-4. A learning rate scheduler with 10%

total training steps and cosine decay was used, with initial and minimum learning rates both 1e-5. The network architecture included a model dimension of 768, 8 transformer layers, and 8 attention heads. Gradient clip value was set to 1.0 and γ to 2.0 in the time-dependent bond-aware noising schedule. For training rectified models, the same hyperparameter setting was applied, except for the loaded pre-trained model checkpoint and the total training epochs set to 10.

Rectification. The trained model applied 100 sampling steps to generate 100 sequences for each peptide length, ranging from 4 to 1035, with a temperature hyperparameter set to 1. After generation, dynamic batching was used to optimize GPU memory utilization for future rectified training.

Evaluation Metrics.

- **Validity** is defined as the fraction of peptide SMILES that pass the SMILES2PEPTIDE filter (Tang et al., 2025b), indicating that it translates to a synthesizable peptide.
- **Uniqueness** is defined as the fraction of mutually distinct peptide SMILES.
- **Diversity** is defined as one minus the average Tanimoto similarity between the Morgan fingerprints of every pair of generated sequences, which measures the similarity in structure across generated peptides.

$$\text{Diversity} = 1 - \frac{1}{\binom{N_{\text{generated}}}{2}} \sum_{i,j} \frac{\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{f}(\mathbf{x}_j)}{|\mathbf{f}(\mathbf{x}_i)| + |\mathbf{f}(\mathbf{x}_j)| - \mathbf{f}(\mathbf{x}_i) \cdot \mathbf{f}(\mathbf{x}_j)}$$

  where $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{f}(\mathbf{x}_j)$ are the 2048-dimensional Morgan fingerprint with radius 3 for a pair of generated sequences $\mathbf{x}_i$ and $\mathbf{x}_j$.
- **Similarity to Nearest Neighbor (SNN)** is defined as the maximum Tanimoto similarity between a generated sequence $\mathbf{x}_i$ with a sequence in the dataset $\tilde{\mathbf{x}}_j$.

$$\text{SNN} = \max_{j \in |\mathcal{D}|} \left(\frac{\mathbf{f}(\mathbf{x}_i) \cdot \mathbf{f}(\tilde{\mathbf{x}}_j)}{|\mathbf{f}(\mathbf{x}_i)| + |\mathbf{f}(\tilde{\mathbf{x}}_j)| - \mathbf{f}(\mathbf{x}_i) \cdot \mathbf{f}(\tilde{\mathbf{x}}_j)} \right)$$

G.3 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.

Performance. PepDFM achieved a validation loss of 3.1051. Its low generalized KL loss during evaluation demonstrates PepDFM's strong capability to generate sequences with high biological plausibility (Gat et al., 2024).

H OBJECTIVE DESCRIPTION

In this work, five key property objectives are considered in the peptide binder tasks: hemolysis, non-fouling, solubility, half-life, and binding affinity. Each of these properties plays a crucial role in optimizing the therapeutic potential of peptides. Hemolysis refers to the peptide's ability to

minimize red blood cell lysis, ensuring safe systemic circulation (Pirtskhalava et al., 2013). Non-fouling properties describe the peptide's resistance to unwanted interactions with biomolecules, thus enhancing its stability and bioavailability in vivo (Chen et al., 2009). Solubility is critical for ensuring adequate peptide dissolution in biological fluids, directly influencing its absorption and therapeutic efficacy (Fosgerau & Hoffmann, 2015). Half-life indicates the duration for which the peptide remains active in circulation, which is vital for reducing dosing frequency (Swanson, 2014). Finally, binding affinity measures the strength of the peptide's interaction with its target, directly correlating to its biological activity and potency in therapeutic applications (Bostrom et al., 2008).

I Score Model Details

We applied the score models from Tang et al. (2025b) to guide the generation of chemically-modified peptide binders. We now introduce the score model developed for the wild-type peptide binder generation task. 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 the ESM-2 protein language model (Lin et al., 2023).

I.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 S1), 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 respectively.

Table S1: 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`

I.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 were 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.

I.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 of 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.

J SAMPLING DETAILS

Score Model Settings. We 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, hemolysis, non-fouling, solubility, and binding affinity, we directly employ their model outputs during sampling.

Wild-Type Peptide Binder Generation Task Settings. The total sampling steps are set to 20 multiplied by the binder length. All possible candidate token transitions are evaluated during each sampling step. We applied the same weight for each objective in all wild-type peptide binder generation tasks.

Chemically-Modified Peptide Binder Generation Task Settings. The total sampling steps are set to 128. With a vocabulary size of 586, evaluating all the possible candidate tokens is too computationally intensive. We therefore only evaluated the top 200 candidate tokens during each sampling step. We applied weight 0.7 for binding affinity, and 0.1 for hemolysis, non-fouling, and solubility, respectively. Instead of random initialization, the initial sequences x_0 are sampled from the pre-trained SMILESReDi[1] with 16 generation steps. During generation, AReUReDi rejects any transitions that will make the SMILES sequence an invalid peptide.

Table S2: Training and validation performance of PepReDi over successive rectification rounds. Each row reports the training loss, validation negative log-likelihood (NLL), validation perplexity (PPL), and conditional total correlation (TC). PepReDi without superscript denotes the base model, while PepReDi^1, PepReDi^3, PepReDi^3 indicate the first, second, and third rounds of rectification, respectively.

	Train Loss	**Val NLL**	**Val PPL**	**Conditional TC**
PepReDi	1.6567	1.6458	5.19	10.6027
PepReDi^1	1.6170	1.6101	5.00	12.6250
PepReDi^2	1.5347	1.5238	4.59	11.7279
PepReDi^3	**1.3538**	**1.3548**	**3.88**	**11.2339**

Table S3: Evaluation metrics for the generative quality of peptide SMILES sequences of max token length set to 200. SMILESReDi without superscription denotes the base model, while SMILESReDi^1 refers to the model that has undergone one round of rectification.

Model	**Validity (↑)**	**Uniqueness (↑)**	**Diversity (↑)**	**SNN (↓)**
Data	1.000	1.000	0.885	1.000
PepMDLM	0.450	1.000	0.705	0.513
SMILESReDi	**0.763**	1.000	0.719	0.593
SMILESReDi^1	**0.986**	1.000	0.665	0.579
PepTune	1.000	1.000	0.677	0.486
AReUReDi	1.000	1.000	**0.789**	**0.392**

Table S4: AReUReDi generates wild-type peptide binders for 8 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 AReUReDi-designed binders.

Name	**Binder Length**	**Hemolysis**	**Non-Fouling**	**Solubility**	**Half-Life (h)**	**Affinity**
AMHR2	8	0.9156	0.8613	0.8564	45.73	7.0608
AMHR2	12	0.9384	0.8872	0.8810	52.52	7.2284
AMHR2	16	0.9420	0.8914	0.8755	63.34	7.2533
EWS::FLI1	8	0.9186	0.8630	0.8619	44.77	5.8424
EWS::FLI1	12	0.9345	0.8819	0.8796	59.11	6.2007
EWS::FLI1	16	0.9416	0.8875	0.8807	64.32	6.4195
MYC	8	0.9180	0.8627	0.8627	44.13	6.4082
OX1R	10	0.9302	0.8687	0.8563	50.14	7.1882
DUSP12	9	0.9240	0.8669	0.8633	48.14	6.1276
1B8Q	8	0.9214	0.8680	0.8654	42.63	5.7130
5AZ8	11	0.9293	0.8732	0.8605	58.33	6.2792
7JVS	11	0.9313	0.8840	0.8743	56.49	6.8449

Table S5: AReUReDi outperforms traditional multi-objective optimization algorithms in designing wild-type 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 (h)	Affinity
1B8Q	MOPSO	8.54	0.8934	0.4763	0.4684	4.45	6.0594
	NSGA-III	33.13	0.9138	0.5715	0.5825	7.32	7.2178
	SMS-EMOA	8.21	0.8804	0.3450	0.3511	3.02	5.955
	SPEA2	17.48	0.9181	0.4973	0.5057	4.13	**7.3240**
	PepTune + DPLM	**2.46**	0.8547	0.3085	0.3213	1.17	5.2398
	AReUReDi	55	**0.9214**	**0.8680**	**0.8654**	**22.93**	5.7130
PPP5	MOPSO	11.34	0.9117	0.4711	0.4255	1.77	6.6958
	NSGA-III	37.30	**0.9521**	0.7138	0.7066	2.90	7.3789
	SMS-EMOA	8.43	0.8758	0.4269	0.4334	1.03	6.2854
	SPEA2	19.02	0.9445	0.6221	0.6098	2.61	**7.6253**
	PepTune + DPLM	**4.80**	0.8816	0.2752	0.2636	1.27	5.8454
	AReUReDi	195	0.9412	**0.896**	**0.8832**	**38.28**	6.7186

Table S6: **Adding a sampling constraint greatly improves AReUReDi's performance.** Wild-type binders for two protein targets (PDB 8CN1 and 4EBP2) were generated with or without a sampling constraint using the same number of generation steps. The table reports the average score for each objective, calculated from 100 generated binders per setting. The best score for each objective is highlighted in bold.

Target	Method	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity
8CN1	w/o constraints	0.8650	0.4782	0.4627	2.54	5.2412
	w/ constraints	**0.9213**	**0.8676**	**0.8697**	**44.70**	**5.5143**
4EBP2	w/o constraints	0.8879	0.4288	0.4257	1.8781	5.7132
	w/ constraints	**0.9356**	**0.8767**	**0.8692**	**53.95**	**6.4571**

Table S7: Ablation results for wild-type peptide binder design targeting PDB 7LUL with different guidance settings. For each setting, 100 binders of length 7 were designed.

Guidance Settings			Hemolysis	Solubility	Affinity
Hemolysis	Solubility	Affinity			
✓	✓	✓	0.9389	0.9398	6.2559
×	✓	✓	0.8964	0.9465	6.3272
✓	×	✓	0.9502	0.4013	6.9798
✓	✓	×	0.9535	0.9642	5.2611
×	×	✓	0.8812	0.2877	7.5057
×	✓	×	0.9036	0.9725	5.2449
✓	×	×	0.9802	0.6135	5.0985
×	×	×	0.8431	0.5810	4.8919

Table S8: Ablation results for wild-type peptide binder design targeting PDB CLK1 with different guidance settings. For each setting, 100 binders of length 12 were designed.

Guidance Settings			Non-Fouling	Half-Life (h)	Affinity
Non-Fouling	Half-Life (h)	Affinity			
✓	✓	✓	0.8285	74.04	6.8099
×	✓	✓	0.2902	96.59	7.3906
✓	×	✓	0.9365	1.33	7.2029
✓	✓	×	0.9479	75.68	6.3437
×	×	✓	0.9625	1.23	6.2319
×	✓	×	0.3540	100.00	6.4116
✓	×	×	0.2531	2.96	8.6580
×	×	×	0.4988	1.82	5.4739

Table S9: **Rectification of the base generation model improves AReUReDi's performance.** Wild-type binders for two protein targets (PDB 5AZ8 and AMHR2) were generated using AReUReDi with three different base models: PepDFM, PepReDi (without rectification), and PepReDi^3 (with three rounds of rectification). The table reports the average score for each objective, calculated from 100 generated binders per setting. The best score for each objective is highlighted in bold.

Target	Base Model	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity
5AZ8	PepDFM	0.9296	**0.8867**	**0.8743**	37.30	6.2291
	PepReDi	**0.9326**	0.8759	0.8572	50.16	**6.4391**
	PepReDi^3	0.9293	0.8732	0.8605	**58.33**	6.2792
AMHR2	PepDFM	0.9412	0.8774	0.8612	47.84	7.2373
	PepReDi	0.9127	0.8602	0.8460	50.92	7.0101
	PepReDi^3	**0.9420**	**0.8914**	**0.8755**	**63.34**	**7.2533**

Table S10: **Annealed guidance strength improves AReUReDi's performance.** Wild-type binders for two protein targets (PDB 1DDV and P53) were generated under four guidance schedules: (1) fixed at the minimum strength $\eta_{min} = 1.0$, (2) fixed at the maximum strength $\eta_{max} = 20.0$, (3) fixed at the midpoint $\frac{1}{2}(\eta_{min} + \eta_{max}) = 10.5$, and (4) an annealed schedule where η_t increases from η_{min} to η_{max} over optimization steps. The table reports the average score for each objective, calculated from 100 generated binders per setting. The best score for each objective is highlighted in bold.

Target	Method	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity
1DDV	$\eta = \eta_{min}$	0.9130	0.8575	0.8429	38.70	5.3554
	$\eta = \eta_{max}$	0.9156	0.8512	0.8479	40.27	5.4359
	$\eta = \frac{1}{2}(\eta_{min} + \eta_{max})$	0.9108	0.8641	0.8544	40.43	5.5396
	$\eta_t = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \frac{t}{T-1}$	0.9128	0.8545	**0.8565**	**44.73**	5.4482
P53	$\eta = \eta_{min}$	0.9335	0.8800	0.8706	49.97	6.2538
	$\eta = \eta_{max}$	0.9293	0.8693	0.8657	61.76	6.3043
	$\eta = \frac{1}{2}(\eta_{min} + \eta_{max})$	0.9294	0.8713	0.8653	59.43	6.3060
	$\eta_t = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \frac{t}{T-1}$	**0.9353**	**0.8818**	**0.8785**	**62.83**	**6.3508**

Table S11: **Best-of-N comparison between PepTune+DPLM and AReUReDi under matched wall-clock time.** For each target, PepTune+DPLM is allowed to generate 100 binders while AReUReDi generates only 4 (PDB 1B8Q) or 3 (PPP5). Top-2 sequences from each method were reported. The table reports the average score for each objective.

Target	Method	Rank	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity
1B8Q	PepTune + DPLM	Top 1	0.9323	0.4379	0.3624	9.82	7.0534
		Top 2	0.8718	0.2573	0.2391	38.67	6.5605
	AReUReDi	Top 1	0.8651	0.8638	0.8892	100.00	5.6008
		Top 2	0.9354	0.8567	0.9331	49.25	6.5605
PPP5	PepTune + DPLM	Top 1	0.7984	0.3338	0.2342	80.27	7.6117
		Top 2	0.7901	0.0966	0.1328	100.00	6.7571
	AReUReDi	Top 1	0.9407	0.9378	0.9131	100.00	6.8193
		Top 2	0.9606	0.8750	0.8399	90.16	6.8969

Table S12: **Increasing generation steps improves AReUReDi's performance.** AReUReDi designed 100 generated binders for MYC (12-mer wild-type peptides) and NCAM1 (chemically-modified peptides of length 200) using different numbers of generation steps. The table reports the average score for each objective. Half-life is not optimized for NCAM1 and is indicated by "*".

Target	# Steps	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity	Time
MYC	64	0.9279	0.8571	0.8519	5.49	6.5167	67
	128	0.9301	0.8721	0.8627	16.54	6.5811	131
	256	0.9357	0.8820	0.8740	34.83	6.5293	265
NCAM1	64	0.8801	0.2468	0.7954	*	5.3936	112
	128	0.8840	0.2657	0.8109	*	5.4377	198
	256	0.8900	0.3015	0.8202	*	5.5929	423

Table S13: Ablation results for wild-type peptide binder design targeting CLK1 with different weight vector settings. For each setting, 100 binders of length 12 were designed. The table reports the average score for each objective.

Weight Vectors			Non-Fouling	Half-Life (h)	Affinity
Non-Fouling	Half-Life	Affinity			
0.3	0.3	0.3	0.8285	74.04	6.8099
0.8	0.1	0.1	0.9367	6.94	6.5231
0.1	0.8	0.1	0.5642	85.47	6.3649
0.1	0.1	0.8	0.6698	48.94	7.4922

Table S14: Ablation results for chemically-modified peptide binder design targeting GFAP with different weight vector settings. For each setting, 100 binders of length 200 were designed. The table reports the average score for each objective.

Weight Vectors			Non-Fouling	Solubility	Affinity
Non-Fouling	Solubility	Affinity			
0.3	0.3	0.3	0.2754	0.8169	5.3011
0.8	0.1	0.1	0.3322	0.7528	5.3487
0.1	0.8	0.1	0.2273	0.8327	5.3378
0.1	0.1	0.8	0.2498	0.7910	5.8827

Table S15: **PepReDi provides prior knowledge that helps AReUReDi to generate samples with better multi-objective trade-offs.** 100 wild-type binders were designed for PDB 1B8Q (8-mer) and PPP5 (16-mer), respectively. The table reports the average score for each objective. The best score for each objective is highlighted in bold.

Target	Prior	Hemolysis	Non-Fouling	Solubility	Half-Life (h)	Affinity
1B8Q	Uniform Prior	0.9009	0.8191	0.8049	14.20	5.8432
	PepReDi Prior	**0.9214**	**0.8680**	**0.8654**	**22.93**	**5.7130**
PPP5	Uniform Prior	0.9265	0.8263	0.7993	17.52	6.7122
	PepReDi Prior	**0.9412**	**0.896**	**0.8832**	**38.28**	**6.7186**

Table S16: **SMILESReDi provides prior knowledge that helps AReUReDi to generate samples with better multi-objective trade-offs.** For each setting, 100 chemically-modified binders of length 200 were designed. The table reports the average score for each objective. The best score for each objective is highlighted in bold.

Target	Prior	Hemolysis	Non-Fouling	Solubility	Affinity
TfR	Uniform Prior	0.8652	0.2381	**0.7777**	5.5535
	SMILESReDi Prior	**0.8665**	**0.3234**	0.7408	**6.1271**
GLP1	Uniform Prior	8.3414	0.2123	**0.7777**	7.5731
	SMILESReDi Prior	**0.8743**	**0.3438**	0.7661	**8.3414**

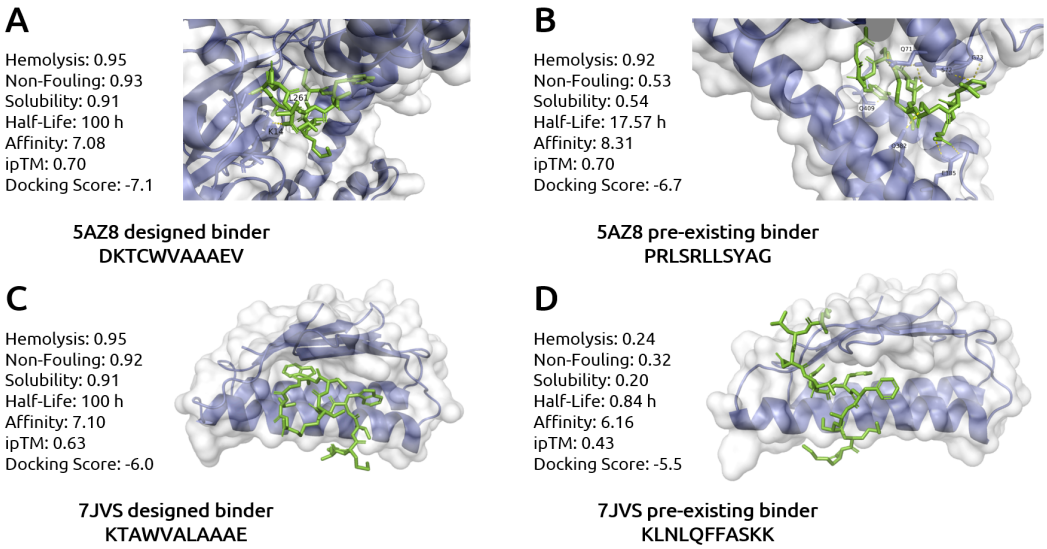

Figure S1: **Complex structures of target proteins with pre-existing binders. (A)-(B)** 5AZ8 **(C)-(D)** 7JVS. Each panel shows the complex structure of the target with either an AReUReDi-designed binder or its pre-existing binder. For each binder, five property scores are provided, as well as the ipTM score from AlphaFold3 and the docking score from AutoDock VINA. Interacting residues on the target are visualized.

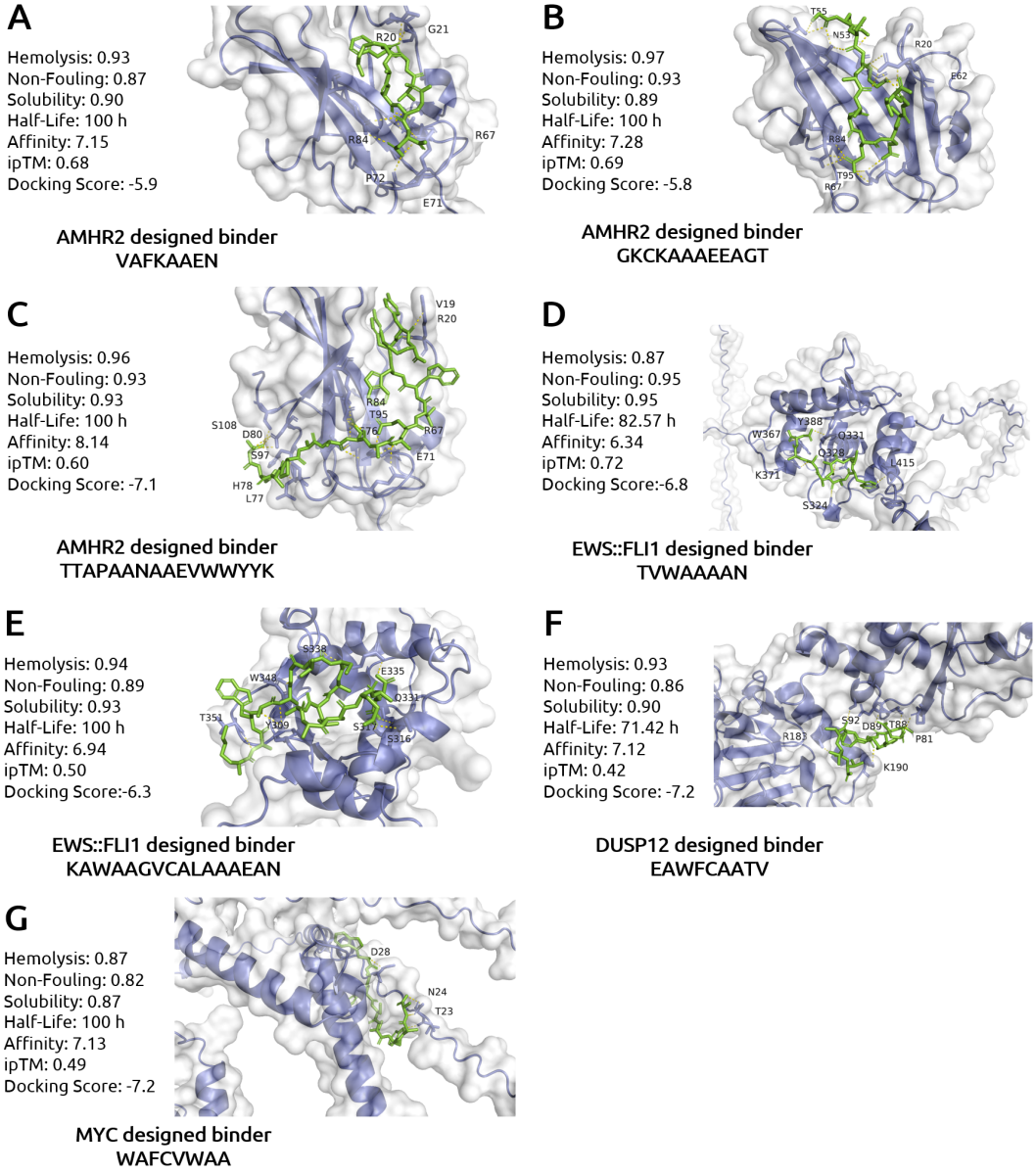

Figure S2: **Complex structures of target proteins without pre-existing binders.** **(A)-(C)** AMHR2, **(D)-(E)** EWS::FLI1, **(F)** MYC, **(G)** DUSP12. Each panel shows the complex structure of the target with an AReUReDi-designed binder. For each binder, five property scores are provided, as well as the ipTM score from AlphaFold3 and the docking score from AutoDock VINA. Interacting residues on the target are visualized.

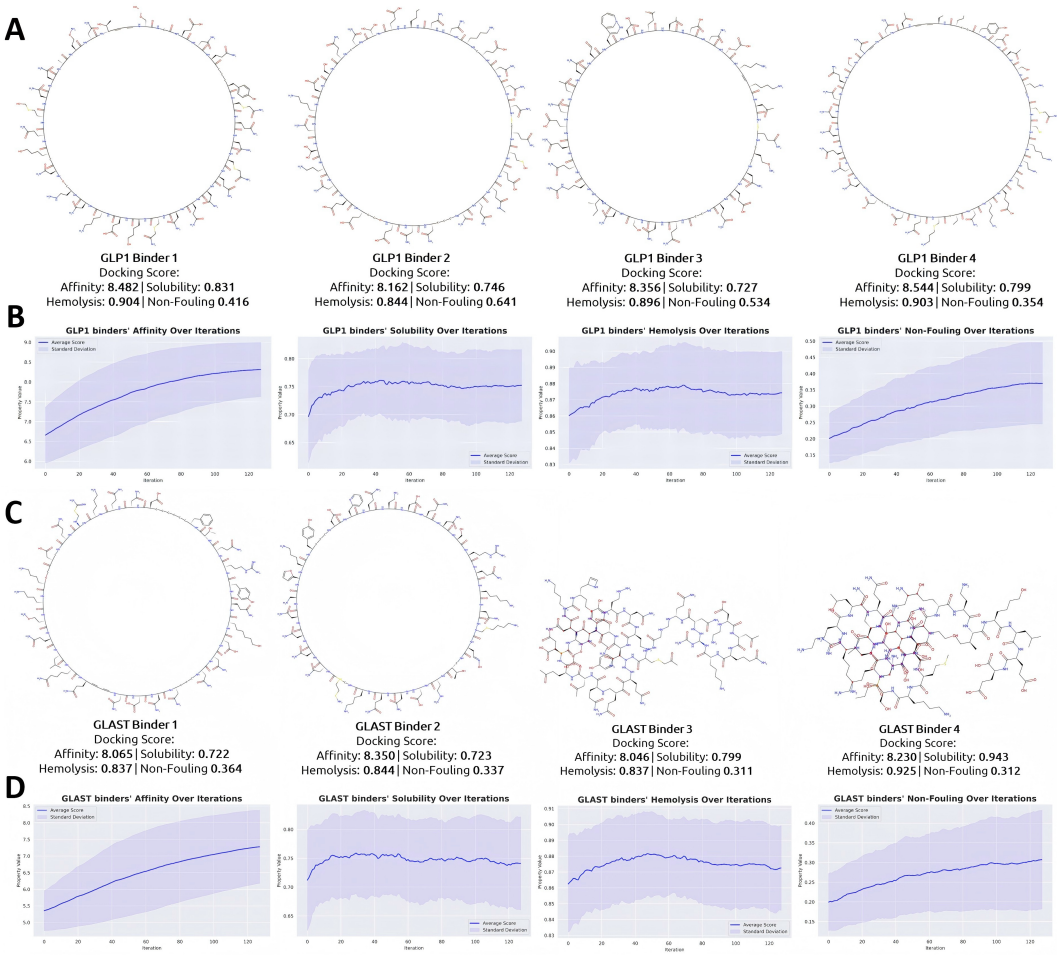

Figure S3: **(A), (C)** Example 2D SMILES structure of AReUReDi-designed peptide binders with four property scores for GLP1 and GLAST, respectively. **(B), (D)** Plots showing the mean scores for each property across the number of iterations during AReUReDi's design of binders of length 200 for GLP1 and GLAST, respectively.

Figure S4: **(A), (C)** Example 2D SMILES structure of AReUReDi-designed peptide binders with four property scores for TfR and AMHR2, respectively. **(B), (D)** Plots showing the mean scores for each property across the number of iterations during AReUReDi's design of binders of length 200 for TfR and AMHR2, respectively.

Algorithm 1 AReUReDi: Annealed Rectified Updates for Refining Discrete Flows

Input: Pre-trained ReDi model $p_t^i(\cdot|x_t)$, objective functions $\tilde{s}_1, \ldots, \tilde{s}_N$, weight vector $\omega \in \Delta^{N-1}$, annealing parameters η_{min}, η_{max}.
Output: Sequence x_T with multi-objective optimized properties.
Initialize:
  Sample an initial sequence x_0 uniformly from the discrete state space S
  Sample or specify a weight vector $\omega \in \Delta^{N-1}$
for $t = 0$ to 1 with step size $h = \frac{1}{T}$ **do**
  Step 1: Annealing and Coordinate Selection
    Update guidance strength: $\eta_t \leftarrow \eta_{min} + (\eta_{max} - \eta_{min})\frac{t}{T-1}$
    Select a position i in the sequence to update: $i \sim \text{Uniform}(\{1, \ldots, L\})$
  Step 2: Proposal Generation via Local Balancing
    Let C_i be the set of candidate tokens from $p_t^i(\cdot|x_t)$.
    For each candidate token $y \in C_i$:
      Compute scalarized reward ratio $r_i(y; x_t)$:

$$r_i(y; x_t) \leftarrow \frac{\exp(\eta_t \min_n \omega_n \tilde{s}_n(x^{(i \leftarrow y)}))}{\exp(\eta_t \min_n \omega_n \tilde{s}_n(x))}$$

      Compute unnormalized proposal distribution $\tilde{q}_i(y|x_t)$ using a balancing function $g(\cdot)$:

$$\tilde{q}_i(y|x_t) \leftarrow p_t^i(y|x_t) g(r_i(y; x_t))$$

    Normalize to get the final proposal distribution $q_i(y|x_t)$.
  Step 3: Metropolis-Hastings Acceptance
    Sample a candidate token $y^* \sim q_i(\cdot|x_t)$.
    Form the proposed state $x_{prop} \leftarrow x^{(i \leftarrow y^*)}$.
    Compute acceptance probability $\alpha_i(x, x_{prop})$:

$$\alpha_i(x, x_{prop}) \leftarrow \min\left\{1, \frac{\pi_{\eta_t,\omega}(x_{prop}) q_i(x^i|x_{prop})}{\pi_{\eta_t,\omega}(x) q_i(y^*|x)}\right\} \quad \pi_{\eta_t,\omega}(z) \propto p_1(z) \exp(\eta_t \min_n \omega_n \tilde{s}_n(z))$$

    With probability $\alpha_i(x, x_{prop})$, accept the proposal: $x \leftarrow x_{prop}$.
    Update time: $t \rightarrow t + h$
end for
Return: Final sequence x_1.

K USE OF LARGE LANGUAGE MODELS (LLMS)

We acknowledge the use of large language models (LLMs) to assist in polishing and editing parts of this manuscript. LLMs were used to refine phrasing, improve clarity, and ensure consistency of style across sections. All technical content, experiments, analyses, and conclusions were developed by the authors, with LLM support limited to language refinement and editorial improvements.