MP2D: Constrained Monte Carlo Tree-Guided Diffusion for Multi-Objective Protein Sequence Design

Zitai Kong[1] , **Yifan Dong**[2] , **Yixuan Wu**[3] , **Zhaokang Liang**[1] , **Jian Wu**[1,3,4*] and **Hongxia Xu**[3*]
[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]School of Mathematical Sciences, Zhejiang University, Hangzhou, China
[3]State Key Laboratory of Transvascular Implantation Devices and TIDRI, Hangzhou, China
[4]Zhejiang Key Laboratory of Medical Imaging Artificial Intelligence, Hangzhou, China
{kongzitai, 3230101152, wyx_chloe, zhaokangliang, wujian2000, einstein}@zju.edu.cn

Abstract

Designing functional protein sequences that satisfy multiple desired properties is a core research focus of protein engineering. Prior methods struggle with inability or inefficiency when dealing with numerous, often conflicting, properties. We propose **Multi-Property Protein Diffusion, (MP2D)**, a unified framework for multi-objective protein sequence optimization that integrates conditional discrete diffusion with constrained MCTS and global iterative refinement. MP2D formulates diffusion denoising as a constrained sequential decision-making process and employs MCTS to explore diverse denoising trajectories guided by Pareto-based rewards. A global iterative refinement strategy further enables repeated remasking and re-optimization of candidate sequences, while a dynamic Pareto constraint prevents candidate bloat and maintains balanced trade-offs across objectives. We evaluate MP2D on two challenging multi-objective protein design tasks: antimicrobial peptide and protein binder optimization, involving four to five conflicting properties. Experimental results demonstrate that MP2D consistently outperforms existing multi-objective baselines, achieving robust and balanced improvements across all objectives without retraining generative models. These results highlight MP2D as a practical and scalable solution for multi-objective functional protein design.

1 Introduction

[1]The design of functional protein sequences often requires the simultaneous optimization of multiple properties, such as activity, stability, toxicity, and specificity. In many real-world applications, including antimicrobial peptides (AMPs) [Wang *et al.*, 2025] and therapeutic protein binders (PBs) [Fleishman *et al.*, 2011], these properties are inherently conflicting [Tokuriki and Tawfik, 2009], making multi-objective optimization a fundamental challenge in protein engineering. Consequently, developing computational methods that can efficiently balance multiple protein properties is critical for practical protein design [Nanda *et al.*, 2017].

Traditional multi-objective optimization approaches typically extend single-property optimization by applying static weighting schemes or hierarchical optimization strategies [Liu *et al.*, 2025c; Yu *et al.*, 2023]. However, such methods are highly sensitive to weight selection and often fail to maintain balanced trade-offs among conflicting objectives [Chen and Li, 2023; Lu *et al.*, 2025]. Pareto-based optimization offers a more principled alternative by explicitly modeling trade-offs among objectives. Nevertheless, as optimization progresses, the Pareto frontier can rapidly bloat and sparsify, which significantly hinders efficient search and makes it difficult to identify high-quality solutions—especially when optimizing more than three or strongly conflicting properties [Giovanelli *et al.*, 2024]. As a result, existing methods struggle to scale to realistic multi-objective protein design problems.

Recent advances in generative models have substantially improved the efficiency of protein sequence design [Alamdari *et al.*, 2023]. To further guide generation toward desirable outcomes, several studies integrate planning algorithms, such as Monte Carlo Tree Search (MCTS), into the inference process of generative models [Liu *et al.*, 2025c]. These training-free approaches enable flexible optimization with plug-and-play reward functions. However, they heavily rely on the initial quality of the generative model and typically perform only single-shot generation, which limits their ability to correct early suboptimal decisions and explore diverse trade-offs [Zhao *et al.*, 2024; Jensen, 2019].

A key observation is that multi-objective protein design inherently admits multiple valid optimization paths, while global property evaluations are often noisy and imperfect. This motivates a search-based and iterative optimization paradigm that can both explore diverse trade-offs and progressively refine candidate solutions. Moreover, effective multi-objective optimization requires explicit mechanisms to control the growth and balance of Pareto candidates.

Here, we propose **Multi-Property Protein Diffusion (MP2D)**, a training-free framework for multi-objective protein sequence optimization. MP2D combines a classifier-free conditional discrete masked diffusion language model for se-

*Corresponding authors.
[1]Appendices are avaliable at https://arxiv.org/abs/2605.05829

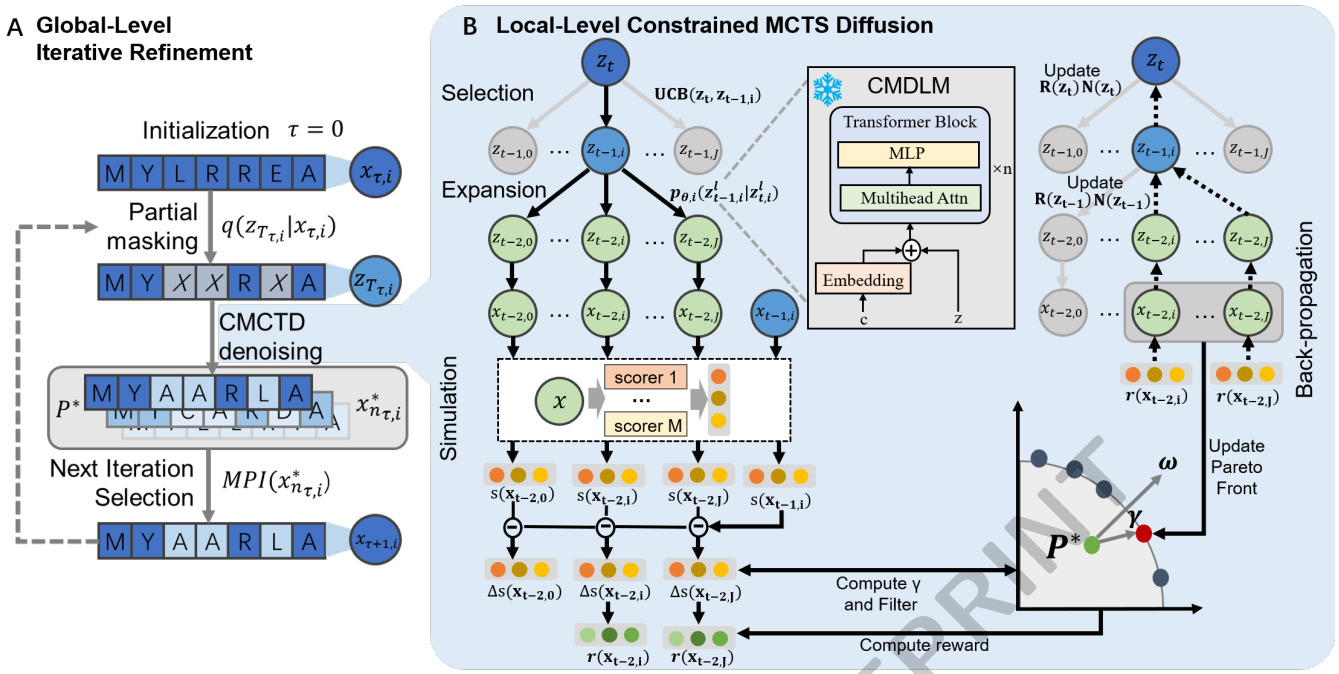

Figure 1: Overview of MP2D. (A) Illustration of global-level iterative refinement process. (B) Visualization of the constrained MCTS-guided diffusion process and the CMDLM model structure.

quence generation with a constrained MCTS-based diffusion refinement process at inference time. A global iterative refinement strategy repeatedly reselects, partially remasks, and re-optimizes candidate sequences, enabling continuous correction of earlier decisions. In addition, MP2D integrates a dynamic constraint on Pareto frontier updates to prevent candidate bloat and maintain balanced optimization across objectives. Our contributions are summarized as follows:

- **A unified design framework for multi-objective protein sequence optimization**. We propose MP2D, a unified framework that balances multiple conflicting protein properties without retraining generative models through constrained MCTS-guided conditional discrete diffusion with global iterative refinement.
- **Conditional masked diffusion language model for protein sequence generation (CMDLM)**. We introduce a classifier-free label-guided conditional masked diffusion language model that provides high-quality, task-specific protein sequence generation as a foundation for optimization.
- **Iterable training-free multi-objective optimization with MCTS**. We propose an inference-time MCTS-guided diffusion framework that supports iterative refinement through global resampling and remasking, enabling robust exploration of multi-objective trade-offs without retraining.
- **Dynamic Pareto frontier constraint for MCTS diffusion (CMCTD)**. We develop a strategy to constrain Pareto frontier updates during MCTS diffusion, preventing optimization collapse caused by frontier bloat.
- **Remarkable empirical performance on functional protein design**. We demonstrate that MP2D effectively balances four to five conflicting properties on antimicrobial peptide and protein binder design tasks, consistently outperforming existing multi-objective baselines.

2 Preliminaries

This section provides a brief overview of the key concepts underlying our approach, including discrete masked diffusion language models, Pareto-based multi-objective optimization, and Monte Carlo Tree Search (MCTS). These preliminaries introduce the notation and standard formulations required for the subsequent sections.

2.1 Discrete Masked Diffusion Language Models

Let $Cat(x; p)$ denote a categorical distribution over a discrete sequence $x = [x_0, x_1, \ldots, x_L]$, parameterized by a probability simplex vector $p \in \mathbf{R}^{|\mathcal{V}|}$. The unconditional masked diffusion framework for discrete sequence generation [Ho *et al.*, 2020] defines a forward noising process $q(x^{(t)}|x^{(t-1)})$ as a Markov process for $t = 0, \ldots, T$, which progressively corrupts the input sequence. The reverse denosing process $p_\theta(x^{(t-1)}|x^{(t)})$ is parameterized by a neural network with parameters θ, and aims to recover less corrupted sequences.

$$q(x^{(t)}|x^{(t-1)}) = Cat(x^{(t)}; \beta_t x^{(t-1)} + (1 - \beta_t) q_{nosie}(x^{(t)})) \quad (1)$$

where q_{nosie} denotes a stationary noise distribution over the vocabulary, and $0 \ll \beta_t < 1$ is a noise schedule controlling the corruption level at diffusion step t. The marginal distribution from the original sequence $x^{(0)}$ admits a closed-form

expression with $\alpha_t = \prod_{i=1}^{t} \beta_i$:

$$q(x^{(t)}|x^{(0)}) = Cat(x^{(t)}; \alpha_t x^{(0)} + (1-\alpha_t) q_{nosie}(x^{(t)}) \quad (2)$$

During inference, new sequences are generated by iteratively applying the learned reverse process

$$p_\theta(x^{(t-1)}|x^{(t)}) = \sum_{\hat{x}_0} q(x^{(t-1)}|x^{(t)}, \hat{x}_0) p_\theta(\hat{x}_0|x^{(t)}) \quad (3)$$

where $\hat{x}_0$ is first sampled from $p_\theta(\cdot|x^{(t)})$ and a less noisy $x^{(t-1)}$ is subsequently sampled by $q(\cdot|x^{(t)}, x^{(0)} = \hat{x}_0)$.

2.2 Pareto Optimization

Optimizing multiple objectives, especially conflicting properties in functional protein design, typically involves severe trade-offs. A widely adopted solution is to identify a set of sequences that cannot be further improved in any single objective without degrading performance in at least one other objective; such sequences are referred to as **Pareto-optimal** [Ngatchou *et al.*, 2005]. Suppose each sequence x is evaluated on M objectives with corresponding score functions, denoted by a vector $s(x) = [s_1(x), \ldots, s_M(x)] \in \mathcal{R}^M$, A sequence x^* is said to dominate another sequence x as

$$\begin{aligned} s(x^*) \succ s(x) \text{ iff } \forall m \in [1, \ldots, M] \text{ s.t. } s_m(x^*) \geq s_m(x) \\ \wedge \exists m' \in [1, \ldots, M] \text{ s.t. } s_{m'}(x^*) > s_{m'}(x) \end{aligned} \quad (4)$$

The set of all sequences that are not dominated by any other sequence is called the **Pareto front**, defined as

$$\mathcal{P}^* = \{x | \nexists x^* \in \mathcal{P}^* \text{ s.t. } s(x^*) \succ s(x)\} \quad (5)$$

2.3 Monte Carlo Tree Search

MCTS [Chaslot *et al.*, 2008] is a planning algorithm that combines tree search with stochastic simulation. It balances exploration and exploitation in the decision space and iteratively refines the search toward high-reward regions. A standard MCTS procedure consists of four stages: selection, expansion, simulation, and backpropagation. Starting from the root node, the selection stage recursively chooses child nodes according to the Upper Confidence Bound (UCB) criterion

$$\text{UCB} = Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}} \quad (6)$$

where Q denotes the estimated reward, c controls the exploration strength, and N represents the visit counts, respectively. In the expansion stage, new child nodes corresponding to previously unvisited actions are added to the tree. The simulation stage performs rollouts from the expanded node to a terminal state to obtain an evaluation. The backpropagation stage updates the Q and N values along the visited path.

3 Methods

In this section, we propose a training-free framework for multi-objective protein sequence optimization based on conditional discrete diffusion, shown in Figure 1. We cast diffusion denoising as a sequential decision-making problem and guided by a dynamically constrained MCTS with Pareto-based rewards. We design a global iterative refinement, which further improves robustness by repeatedly masking and re-optimizing candidate sequences. The implementation details and hyperparameter settings can be found in Appendix A.

3.1 Conditional masked diffusion language model

Directly applying pretrained diffusion language models trained on large-scale, general protein corpora often results in an excessively large search space, which makes subsequent reward-guided generation inefficient and may hinder exploration of valid local protein regions. To address this issue, we extend the unconditional masked diffusion language model for discrete sequences to CMDLM, a label-guided, classifier-free conditional diffusion model tailored to specific functional protein design tasks. We introduce a condition label c, representing the protein type, into the diffusion model and parameterize the reverse process as $p_\theta(x^{(0)}|x^{(t)}, c)$. To enable classifier-free guidance, the model combines conditional and unconditional predictions [Ho and Salimans, 2022] with

$$\hat{l} = (1+w) p_\theta(x^{(0)}|x^{(t)}, c) - w p_\theta(x^{(0)}|x^{(t)}, \emptyset) \quad (7)$$

where $w \geq 0$ controls the guidance strength and $\emptyset$ denotes the unconditional condition. The resulting guided distribution is

$$\tilde{p}_\theta(x^{(0)}|x^{(t)}, c) = softmax(\hat{l}) \propto \frac{p_\theta(x^{(0)}|x^{(t)}, c)^{1+w}}{p_\theta(x^{(0)}|x^{(t)}, \emptyset)^w} \quad (8)$$

In practice, we jointly train the conditional and unconditional models by randomly replacing c with the unconditional identifier $\emptyset$ with probability p_{uncond}. The training objective is a reweighted cross-entropy loss

$$\mathcal{L}_t = \mathbb{E}_{q(x^{(0)})}[\lambda^{(t)} \sum_{i=1}^{L} b_i(t) \log \tilde{p}_\theta(x_i^{(0)}|x^{(t)}, c)] \quad (9)$$

where $\lambda^{(t)}$ is a timestep-dependent weight induced by the noise schedule, and $b_i(t) = \mathbb{I}[x_i^{(t)} \neq x_i^{(0)}]$ indicates corrupted positions. During generation, the initial sequence is fully masked. At each diffusion step, a subset of masked tokens is updated by sampling

$$p_\theta(x^{(t-1)}|x^{(t)}) = \sum_{\hat{x}_0} q(x^{(t-1)}|x^{(t)}, \hat{x}_0) \tilde{p}_\theta(\hat{x}_0|x^{(t)}, c) \quad (10)$$

This conditional model serves as the backbone for our subsequent constrained MCTS-based diffusion framework.

3.2 Constrained MCTS Diffusion

Given the conditional diffusion backbone, we formulate the denoising process as a sequential decision-making problem and employ MCTS to explore multiple denoising trajectories under flexible multi-objective constraints. The proposed framework relies exclusively on pretrained models and external property evaluators, requiring no additional training while enabling scalable optimization over an arbitrary number of objectives through constraint-based filtering. By integrating global objective evaluation with local diffusion transitions, this module supports efficient exploration of the constrained protein design space. Detailed algorithm is in Appendix C.4

Initialization Let z_t denote a partially unmasked sequence at diffusion step t, and let $child(z_t) = \{z_{t-1,0}, \ldots, z_{t-1,J}\}$ denote its child nodes generated by one-step denoising. To enable global iterative refinement from candidate proteins, we initialize the MCTS root as a partially masked sequence

$z_T \in |\mathcal{V}|^L$ with noise level T. The Pareto set $\mathcal{P}^*$ is initialized as empty. We assume M pretrained or selected scoring functions $\{s_m(\cdot)\}_{m=1}^M$. To control optimization trade-offs, we construct a set of direction vectors $\omega \in \mathbf{R}^M$ using the Das–Dennis simplex lattice with H subdivisions:

$$\omega_i = \frac{q_i}{H}, \text{ where } q_i \in \mathbf{Z}_{\geq 0}, \sum_{i=1}^{M} q_i = H \tag{11}$$

Selection Starting from the root, the tree is traversed until a leaf node is reached. At each intermediate node z_t, we restrict candidate actions to its Pareto non-dominated children:

$$\begin{aligned}\mathcal{P}^*_{z_t} =&\{z_{t-1,i} | \nexists z_{t-1,j} \in \text{child}(z_t) \\ &\text{s.t. } UCB(z_t, z_{t-1,j}) \succ UCB(z_t, z_{t-1,i})\}\end{aligned} \tag{12}$$

One child is uniformly sampled from $\mathcal{P}^*_{z_t}$ and the modified UCB score is defined as

$$UCB(z_t, z_{t-1,i}) = \frac{R(z_{t-1,i})}{N(z_{t-1,i})} + cp_\theta(z_{t-1,i}|z_t)\frac{\sqrt{N(z_t)}}{1+N(z_{t-1,i})} \tag{13}$$

where $R(\cdot)$ and $N(\cdot)$ denote the cumulative reward vector and visit count, respectively. The first term promotes exploitation of high-reward nodes, while the second term encourages exploration guided by the diffusion posterior to ensure validity.

Expansion For a selected leaf node z_t that is not fully unmasked, we generate J child nodes z_{t-1} by sampling from the diffusion model $p_\theta(z_{t-1,i}|z_t)$. To avoid duplicate generations, we inject independent Gumbel noise into the token-level logits [Tang *et al.*, 2025]:

$$p_{\theta,i}(z^l_{t-1,i}|z^l_t) = \log p_\theta(z^l_{t-1,i}|z^l_t) + \mathcal{G}^l_i \tag{14}$$

where $\mathcal{G}^l_i = -\log(-log(u^l_{i,j}+\epsilon)+\epsilon)$, $u^l_{i,j} \sim \text{Uniform}(0,1)$.

Simulation and Constraint Filtering Each child node $z_{t-1,i}$ is fully denoised via greedy decoding to obtain a clean sequence $x_{t-1,i}$, which is evaluated by all scoring functions to yield $s(x_{t-1,i}) = [s_1(x_{t-1,i}), \ldots, s_M(x_{t-1,i})]$.

To prevent rapid expansion of the search tree and uncontrolled growth of the Pareto set, we constrain candidate additions using predefined optimization directions. We compute the improvement vector

$$\Delta s_m(x_{t-1,i}) = s_m(x_{t-1,i}) - s_m(x_t) \tag{15}$$

and its cosine similarity with direction ω:

$$\gamma(z_{t-1,i}) = \frac{\Delta s(x_{t-1,i}) \cdot \omega}{\|\Delta s(x_{t-1,i})\| \, \|\omega\|} \tag{16}$$

Only candidates satisfying $arccos(\gamma) \leq \Psi$ are retained; fallback rules are applied when no such candidates exist (see Appendix C.1). To avoid excessive rejection or overly permissive filtering during different stages of the search, the angular threshold Ψ can be adaptively adjusted to maintain a stable acceptance rate (see Appendix C.2).

For each retained sequence, the reward vector $r(x_{t-1,i}) = [r_1(x_{t-1,i}), \ldots, r_M(x_{t-1,i})] \in \mathbf{R}^M$ is computed as

$$r_m(x_{t-1,i}) = \frac{1}{|\mathcal{P}^*|}\sum_{n=1}^{|\mathcal{P}^*|} \mathbf{1}[s_m(x_{t-1,i}) \geq s_m(x^*_n)] \tag{17}$$

indicating its relative dominance over current Pareto candidates. Finally, the Pareto set is updated by adding non-dominated sequences and removing dominated ones

$$\mathcal{P}^* = \mathcal{P}^* \cup \{x_{t-1,i} | \forall x^*_n \in \mathcal{P}^* \; s(x_{t-1,i}) \succeq s(x^*_n)\} \tag{18}$$

$$\mathcal{P}^* = \mathcal{P}^* / \{x^*_n | \exists x_{t-1,i} \text{ s.t. } s(x_{t-1,i}) \succeq s(x^*_n)\} \tag{19}$$

Back-propagation For a newly explored node $z_{t-1,i}$, we update the cumulative reward vector as $R(z_{t-1,i}) = r(x_{t-1,i})$ and its visit count as $N(z_{t-1,i}) = 1$. For each ancestor node z_t along the path to the root, we update

$$R(z_t) = R(z_t) + \sum_{i=1}^{J} r(x_{t-1,i}) \tag{20}$$

$$N(z_t) = N(z_t) + 1 \tag{21}$$

3.3 Global-Level Iterative Refinement

Although a single-shot CMCTD step can produce improved sequences, it still faces two limitations. First, global property evaluators may introduce noisy or inconsistent assessments, and once a token is unmasked, the decision cannot be revised within the same denoising trajectory. This makes single-pass optimization vulnerable to early errors. Second, different masking patterns can lead to different optimization paths, and relying on a single denoising trajectory reduces candidate diversity and may miss better solutions. Detailed algorithm can be found in Appendix C.3.

To address these issues, we introduce a global iterative refinement framework, where sequences are progressively improved through multiple cycles of partial masking and constrained MCTS-based denoising. This enables more robust optimization of global properties and allows continual correction of earlier decisions.

Initialization Let the refinement process start with W seed protein sequences $\{x_{\tau,1}, \ldots, x_{\tau,W}\} \in \mathcal{V}^{W \times L}$, with $\tau = 0$, where $\tau \in [0, \tau_{max}]$ denotes the refinement iteration. At each iteration, a noise level $T_{\tau,i}$ is sampled for each sequence.

Partial Masking For every sequence $x_{\tau,i}$, we construct a partially masked sequence $z_{T_{\tau,i}}$ with

$$q(z_{T_{\tau,i}}|x_{\tau,i}) = Cat(z_{T_{\tau,i}}; \alpha_{T_{\tau,i}}x_{\tau,i} + (1-\alpha_{T_{\tau,i}})q_{noise}(z_{T_{\tau,i}}) \tag{22}$$

where $\alpha_{T_{\tau,i}} = \prod_{i=1}^{T_{\tau,i}} \beta_i$. This masked sequence serves as the starting point for constrained MCTS diffusion.

MCTS-Based Denoising A single constrained MCTS diffusion step is applied to each partially masked sequence, producing a set of Pareto non-dominated candidates $\mathcal{P}^*$.

Next Iteration Selection Despite the constraint mechanism, the Pareto set may still grow large ($|\mathcal{P}^*| > W$). To select a representative next-iteration seed, we evaluate each candidate $x^*_{n,\tau,i} \in \mathcal{P}^*$ by computing the improvement vector $\Delta s(x^*_{n,\tau,i})$ with

$$\Delta s_m(x^*_{n,\tau,i}) = s_m(x^*_{n,\tau,i}) - s_m(x_{\tau,i}) \tag{23}$$

We then define a multi-property improvement (MPI) score

$$\begin{aligned}MPI(x^*_{n,\tau,i}) = \text{z-score}(&\frac{1}{M}\sum_{m=1}^{M} \mu_m \text{Rank}_m(x^*_{n,\tau,i})) \\ &+ \lambda \text{z-score}(D(x^*_{n,\tau,i}))\end{aligned} \tag{24}$$

Methods	Uniprot Peptide			Protein Binder			AMP		
	pLDDT(↑)	pppl(↓)	FPD(↓)	pLDDT(↑)	pppl(↓)	FPD(↓)	pLDDT(↑)	pppl(↓)	FPD(↓)
dataset	71.35	12.57	0.2317	71.45	12.58	0.1961	73.69	12.05	0.2617
ProteinGAN	67.84	15.13	1.5067	65.15	15.90	2.3336	60.23	14.37	2.7669
ProtGPT2	70.86	12.68	3.3646	70.14	13.39	3.1198	70.61	**13.19**	3.5444
EvoDiff	64.99	14.25	0.6011	68.96	14.39	0.5944	69.68	13.51	0.6352
CMDLM	**73.12**	**11.44**	**0.3280**	**70.68**	**13.32**	**0.4106**	**71.84**	13.33	**0.5423**

Table 1: Performance comparison between CMDLM and baseline models on three tasks: Uniprot peptides, protein binders and AMPs.

where

$$\text{Rank}_m(x^*_{n,\tau,i}) = \frac{rank(\Delta s_m(x^*_{n,\tau,i}))}{T_{\tau,i}} \tag{25}$$

$$D(x^*_{n,\tau,i}) = \Delta s(x^*_{n,\tau,i}) \cdot \omega \tag{26}$$

Here, μ is normalized weights ($\sum_{m=1}^{M} \mu_m = 1$), $\text{Rank}_m(\cdot)$ measures the relative improvement on each objective, where the $T_{\tau,i}$ division is to ensure comparability across different corruption levels. $D(\cdot)$ measures alignment with the predefined optimization direction. Finally, the next-iteration sequence is sampled from the Pareto set according to:

$$x_{\tau+1,i} \sim \frac{\exp MPI(x^*_{n,\tau,i})}{\sum_{x\in\mathcal{P}^*} \exp MPI(x)} \tag{27}$$

4 Experiments

We pretrained CMDLM on general peptides and performed conditional finetunes and multi-property optimizations on two important protein medicines, protein binders (PBs) and antimicrobial peptides (AMPs). In this section, we first discuss the benchmarks used. Then we introduce and analysis the performance of the CMDLM generation and the CMCTD optimization on both cases. Additional visualization analysis can be found in Appendix D.

4.1 Experimental Setup

Datasets

For pretraining CMDLM, we collected peptide sequences of length 2–50 from UniProt [Consortium, 2019], yielding 2.6M sequences after removing those containing uncommon amino acids. For antimicrobial peptide (AMP) design, we followed the data collection procedure in [Chen *et al.*, 2024] and combined sequences from dbAMP [Yao *et al.*, 2025], AMP Scanner [Veltri *et al.*, 2018], and DRAMP [Shi *et al.*, 2022]. After filtering for length (< 40) and standard amino acids, we obtained 195k peptides. For protein binder (PB) design, we adopted the dataset from [Chen *et al.*, 2025], which includes 15.5k peptides curated from PepNN, BioLip2, and PPIRef. Sequences range from 6 to 49 residues, and those with uncommon amino acids were removed. All datasets are splited into training and validation sets with a ratio of 9:1.

Baselines

To evaluate CMDLM, we followed the benchmark protocol of [Meshchaninov *et al.*, 2024], comparing against: ProteinGAN [Repecka *et al.*, 2021] (GAN-based), EvoDiff [Alamdari *et al.*, 2023] (discrete diffusion), ProtGPT2 [Ferruz *et al.*, 2022] (autoregressive). All baseline models were configured with comparable parameter sizes and standard character-level tokenization to ensure fairness.

For the PB optimization task, we followed the benchmark of [Chen *et al.*, 2025], including four classical multi-objective evolutionary algorithms—NSGA-III, SMS-EMOA, SPEA2, and MOPSO—and a recent discrete flow matching–based method MOG-DFM, specifically introduced for protein binder optimization.

To the best of our knowledge, no established benchmark exists for multi-objective AMP optimization. We therefore constructed one using advanced AMP design models that are reported to be able to optimize over 3 properties, including Multi-CGAN [Yu *et al.*, 2023], MPOGAN [Liu *et al.*, 2025a], HMAMP [Wang *et al.*, 2025], and the multimodal fusion model MoFormer [Wang *et al.*, 2024].

Detailed descriptions and selection reasons of baseline methods can be found in Appendix E.

Metrics

Foundation Model Evaluation We used three metrics to assess CMDLM's performance: **ESM-2 perplexity** [Lin *et al.*, 2022] for sequence plausibility, **pLDDT** [Jumper *et al.*, 2021] for structural foldability, and **Frechet ProtT5 Distance (FPD)** for distributional similarity.

Protein Binder Optimization We evaluated five therapeutically relevant properties: **hemolysis**, **non-fouling**, **solubility**, **half-life**, and **binding affinity**. Predictions were made using the public classifiers provided in [Liu *et al.*, 2025a].

AMP Optimization We used publicly available predictors to evaluate key properties: AMP-Scanner [Veltri *et al.*, 2018] for **antimicrobial probability**, HemoPI [Chaudhary *et al.*, 2016] for **hemolysis**, and ToxinPred2 [Rathore *et al.*, 2024] for **toxicity**. we also evaluated **MIC values** using a regressor trained on MIC data collected from GRAMPA [Plisson, 2022] and DBAASP [Pirtskhalava *et al.*, 2021] (see Appendix B). In addition, detailed definitions and selection criteria can be found in Appendix F.

4.2 Evaluation of Conditional Diffusion Backbone

As the generative backbone for subsequent optimization, CMDLM is expected to capture class-specific sequence patterns while producing valid and realistic protein sequences. Due to the limited availability of labeled AMP and protein binder data, we first pretrain the model unconditionally on large-scale peptide sequences to learn general peptide syntax, and then fine-tune it with LoRA under conditional supervision to specialize in target protein classes.

Target	Methods	Hemolysis(↓)	Non-Fouling(↑)	Solubility(↑)	Half-Life(↑)	Affinity(↑)
1B8Q	MOPSO†	0.1066	0.4763	0.4684	4.449	6.0594
	NSGA-III†	0.0862	0.5715	0.5825	7.324	7.2178
	SMS-EMOA†	0.1196	0.3450	0.3511	3.023	5.9550
	SPEA2†	0.0819	0.4973	0.5057	4.126	**7.3240**
	MOG-DFM†	0.0785	0.8445	0.8455	27.227	5.9094
	MP2D	**0.0460**	**0.8928**	**0.8977**	**29.032**	6.7794
PPP5	MOPSO†	0.0883	0.4711	0.4255	1.769	6.6958
	NSGA-III†	0.0479	0.7138	0.7066	2.901	7.3789
	SMS-EMOA†	0.1242	0.4269	0.4334	1.031	6.2854
	SPEA2†	0.0555	0.6221	0.6098	2.613	**7.6253**
	MOG-DFM†	0.0617	0.7738	0.7510	27.775	6.8197
	MP2D	**0.0472**	**0.8900**	**0.8866**	**41.5726**	7.2007

Table 2: Performance on protein binder design of MP2D and baseline methods. †: benchmark results quoted from [Liu *et al.*, 2025a].

We evaluate CMDLM from three complementary aspects: sequence plausibility, structural foldability, and distributional alignment. Sequence quality is assessed using ESM-2 perplexity, which measures consistency with natural protein statistics. Structural plausibility is evaluated by the predicted local distance difference test (pLDDT) score. Distributional similarity is measured via the Frechet ProtT5 Distance (FPD) between generated sequences and real proteins from the corresponding classes. Across all evaluation settings, CMDLM almost consistently achieves lower ESM-2 perplexity and FPD, together with higher pLDDT, compared to baseline methods. These results indicate that CMDLM generates sequences that are statistically reliable, structurally foldable, and well aligned with natural protein distributions, making it a strong foundation for downstream optimization through diffusion-based denoising.

4.3 Evaluation of Multi-objective Optimization

MP2D Generates PBs Satisfying Multiple Properties

Following the benchmark of [Liu *et al.*, 2025a], we designed peptide binders for two target proteins: 1B8Q, a smaller protein with a known binder, and PPP5, a larger protein without characterized binders. We first sampled 100 binder candidates of random lengths from CMDLM conditioned on the PB label, and optimized five key therapeutic properties—hemolysis, non-fouling, solubility, half-life, and binding affinity—for 100 refinement iterations. Each baseline method generated the same number of candidates per target for a fair comparison. As shown in Table 2, MP2D achieves the lowest hemolysis, the highest non-fouling, solubility, half-life, and competitive binding affinity, consistently outperforming all baselines. Figure S2 shows that MP2D-designed binders surpass natural ones across all properties.

Importantly, classical baseline methods are unable to optimize all five conflicting properties simultaneously, often collapsing one or more properties when improving others. In contrast, MP2D remains stable across all objectives, demonstrating its ability to navigate complex trade-offs and maintain balanced improvements.

MP2D Designs Optimized AMPs on Scalar Properties

We performed a parallel procedure for AMP optimization. CMDLM generated 100 candidate peptides under the AMP condition with random lengths, which were then optimized jointly for four critical properties: antimicrobial activity, MIC value, hemolysis, and toxicity. Baseline models were used to generate 100 AMP candidates under the same settings. As summarized in Table 3, MP2D produces AMPs with the highest antimicrobial probability and the lowest MIC, hemolysis and toxicity. Consistently with the binder case, MP2D-designed AMPs also outperform natural AMPs across all four properties (Figure S2).

Method	Pamp(↑)	MIC(↓)	Hemolysis(↓)	Toxicity(↓)
Multi-CGAN	0.3785	1.3214	0.5813	0.7698
MPOGAN	0.6201	1.2346	0.6107	0.6166
MoFormer	0.4326	1.3398	0.5761	0.8198
HMAMP	0.6714	1.4266	0.5613	0.7641
MP2D	**0.9949**	**0.2689**	**0.4077**	**0.2968**

Table 3: Performance on AMP of MP2D and baseline methods.

Across baselines, we again observe that no existing method can simultaneously improve all four conflicting AMP properties, often leading to performance degradation in at least one objective. MP2D avoids this collapse and produces balanced improvements across all objectives, underscoring its robustness in handling strongly conflicting biological properties.

Moreover, MP2D is training-free and plug-and-play during optimization. By simply replacing the property predictor, MP2D can be directly applied without collecting additional training data or retraining any model—highlighting its practicality and broad applicability.

4.4 Ablation Studies

We conduct ablation studies on both protein binder and AMP tasks to examine the contribution of each component in MP2D. Results are summarized in Table 4 and Table 5.

Conditional Base Model Narrows the Optimization Gap

We first analyze the impact of initialization strategies and conditional diffusion. Using random peptides as starting points together with the unconditional diffusion model pretrained on general peptides yields the weakest performance across all objectives. Initializing from AMP or protein binder sequences improves optimization quality, indicating that reasonable starting points can partially facilitate search.

In contrast, replacing the unconditional diffusion model with a conditional one leads to substantially larger improvements, even when starting from generic peptides. This demonstrates that conditional diffusion plays a more critical role than initialization alone by constraining the search space to task-relevant protein distributions. Combining AMP or PB initialization with conditional diffusion achieves the best performance, highlighting their complementary effects.

Constrained Optimization Prevents Property Collapse

To evaluate the role of constraint filtering, we remove this component while keeping all other modules unchanged. Without constraint filtering, optimization performance degrades noticeably, and several objectives deteriorate despite improvements in others. This indicates that constraint filter-

Target	start w/ PBs	conditional model	constraint	refined	Hemolysis (↓)	Non-Fouling (↑)	Solubility (↑)	Half-Life (↑)	Affinity (↑)
1B8Q			✓	✓	0.1313	0.4660	0.4723	9.2736	5.4521
		✓	✓	✓	0.0821	0.7174	0.7130	19.4659	5.9339
	✓		✓	✓	0.0940	0.6442	0.6416	12.7857	5.6422
	✓	✓		✓	0.0460	0.8928	0.8977	15.0126	6.1130
	✓	✓	✓		0.1289	0.3377	0.3585	3.4126	5.2340
PPP5			✓	✓	0.1290	0.4575	0.4654	8.2453	5.9767
		✓	✓	✓	0.0823	0.7155	0.7174	18.0461	6.4319
	✓		✓	✓	0.1089	0.5889	0.5995	7.8438	6.0806
	✓	✓		✓	0.0590	0.8681	0.8684	27.3747	6.7360
	✓	✓	✓		0.1138	0.4102	0.4226	2.0140	5.0023

Table 4: Ablation study results on protein binder design task.

start w/ AMPs	conditional model	constrained	refined	Pamp(↑)	MIC(↓)	Hemolysis(↓)	Toxicity(↓)
		✓	✓	0.3302	0.9842	0.4466	0.0064
	✓	✓	✓	0.6104	0.9035	0.4274	0.1415
✓		✓	✓	0.7986	0.9526	0.4746	0.4012
✓	✓		✓	0.9712	0.6533	0.5400	0.3319
✓	✓	✓		0.8489	0.7749	0.5244	0.4774

Table 5: Ablation study results on the AMP design task.

ing is essential for preventing Pareto front collapse and maintaining balanced trade-offs among conflicting properties.

Iterative Refinement Gives Further Improvements
Finally, we assess the effect of global iterative refinement by disabling the remasking-and-reoptimization process. Without iterative refinement, optimization quickly saturates and fails to correct early suboptimal decisions. As a result, both AMP and PB tasks exhibit inferior overall performance compared to the full MP2D framework. These results confirm that global iterative refinement is crucial for robust multi-objective optimization under noisy global evaluations.

5 Related Work

Generation models for *de novo* protein sequence design aim to produce novel protein sequences from scratch. Recent advances have demonstrated promising results using autoregressive models such as ProtGPT2 [Ferruz *et al.*, 2022], GAN-based approaches such as ProteinGAN [Repecka *et al.*, 2021], and diffusion-based models such as EvoDiff [Alamdari *et al.*, 2023]. These models are effective at capturing the statistical patterns of natural protein sequences and generating realistic candidates. However, they do not explicitly optimize sequences toward multiple objectives during inference. As a result, additional optimization mechanisms are often required to steer generation toward desired property trade-offs.

Planning-based generation models processes toward high-quality outputs under task-specific objectives. MCTS and related lookahead techniques are widely used general mechanism. RethinkMCTS [Li *et al.*, 2025] applies MCTS to refine reasoning steps in LLMs, while MCTD [Yoon *et al.*, 2025] integrates diffusion denoising with tree search. In biomolecular design, ProtInvTree [Liu *et al.*, 2025b] and PepTune [Tang *et al.*, 2025] similarly combine tree search with denoising processes for inverse folding and molecular design tasks. However, these methods can become inefficient due to the combinatorial growth of candidate expansions and the accumulation of weakly non-dominated solutions when extended to numerous objectives.

Multi-objective protein sequence design seeks sequences that satisfy several desired, and often conflicting, properties. A common strategy is scalarization, which reduces multiple objectives to a single score via weighted sums or hierarchical priorities. MCTD-ME [Liu *et al.*, 2025c] extends MCTD by aggregating multiple properties into a weighted evaluation score. Multi-CGAN [Yu *et al.*, 2023] conditions generative models on combinations of property labels, while HMAMP [Wang *et al.*, 2025] employs multiple discriminators to enforce property constraints during training. Another paradigm performs model-based sequence optimization using learned predictors. MPOGAN [Liu *et al.*, 2025a] iteratively retrains generators with classifier-filtered samples, and MOG-DFM [Chen *et al.*, 2025] guides flow matching with adaptive probability paths for multi-objective optimization. These methods typically rely on repeated retraining or careful tuning of optimization signals, and may struggle to maintain balanced trade-offs as the number of objectives increases.

Iterative training-free optimization focuses on progressively refining candidate solutions without retraining underlying models. Classical population-based methods, including swarm-based approaches such as MOPSO [Coello and Lechuga, 2002] and evolutionary algorithms such as NSGA-III [Ishibuchi *et al.*, 2016], SMS-EMOA [Beume *et al.*, 2007], and SPEA2 [Zitzler *et al.*, 2001], exemplify this paradigm through repeated selection and variation. These methods have long been used as generic multi-objective optimizers. In modern protein design pipelines, similar iterative schemes are combined with deep generative models. RERD [Uehara *et al.*, 2025] repeatedly denoises protein sequences along a single optimization direction. Nevertheless,integrating iterative search with conditional diffusion under Pareto constraints remains underexplored for multi-objective optimization.

6 Conclusions

In this paper, we presented MP2D, a unified framework for multi-objective protein sequence generation and optimization that combines conditional diffusion with constrained MCTS and iterative refinement. By explicitly controlling Pareto candidate evolution and enabling iterative correction during inference, MP2D effectively addresses the challenges of optimizing multiple conflicting properties. Experiments on AMP and PB design tasks show that MP2D consistently achieves balanced improvements across multiple objectives and outperforms existing task-specific multi-objective baselines without retraining generative models. In future work, MP2D can be extended to other protein design tasks. We believe this work provides a practical step toward scalable and reliable protein engineering.

Acknowledgements

This research was partially supported by Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China No. JYB2025XDXM601, Na-

tional Natural Science Foundation of China under Grant No. T2541004 and TIDRI under Grant No. KY052025003.

Contribution Statement

Zitai Kong and Yifan Dong contributed equally to this study.

References

[Alamdari *et al.*, 2023] Sarah Alamdari, Nitya Thakkar, Rianne Van Den Berg, Neil Tenenholtz, Robert Strome, Alan M Moses, Alex X Lu, Nicolò Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pages 2023–09, 2023.

[Beume *et al.*, 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.

[Chaslot *et al.*, 2008] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. Monte-carlo tree search: A new framework for game ai. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 4, pages 216–217, 2008.

[Chaudhary *et al.*, 2016] Kumardeep Chaudhary, Ritesh Kumar, Sandeep Singh, Abhishek Tuknait, Ankur Gautam, Deepika Mathur, Priya Anand, Grish C Varshney, and Gajendra PS Raghava. A web server and mobile app for computing hemolytic potency of peptides. *Scientific reports*, 6(1):22843, 2016.

[Chen and Li, 2023] Tao Chen and Miqing Li. The weights can be harmful: Pareto search versus weighted search in multi-objective search-based software engineering. *ACM Transactions on Software Engineering and Methodology*, 32(1):1–40, 2023.

[Chen *et al.*, 2024] Tianlai Chen, Pranay Vure, Rishab Pulugurta, and Pranam Chatterjee. Amp-diffusion: Integrating latent diffusion with protein language models for antimicrobial peptide generation. *bioRxiv*, pages 2024–03, 2024.

[Chen *et al.*, 2025] Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. *arXiv preprint arXiv:2505.07086*, 2025.

[Coello and Lechuga, 2002] 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, pages 1051–1056. IEEE, 2002.

[Consortium, 2019] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[Ferruz *et al.*, 2022] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[Fleishman *et al.*, 2011] Sarel J Fleishman, Timothy A Whitehead, Damian C Ekiert, Cyrille Dreyfus, Jacob E Corn, Eva-Maria Strauch, Ian A Wilson, and David Baker. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science*, 332(6031):816–821, 2011.

[Giovanelli *et al.*, 2024] Joseph Giovanelli, Alexander Tornede, Tanja Tornede, and Marius Lindauer. Interactive hyperparameter optimization in multi-objective problems via preference learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12172–12180, 2024.

[Ho and Salimans, 2022] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[Ishibuchi *et al.*, 2016] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. Performance comparison of nsga-ii and nsga-iii on various many-objective test problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3045–3052. IEEE, 2016.

[Jensen, 2019] Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

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

[Li *et al.*, 2025] Qingyao Li, Wei Xia, Xinyi Dai, Kounianhua Du, Weiwen Liu, Yasheng Wang, Ruiming Tang, Yong Yu, and Weinan Zhang. Rethinkmcts: Refining erroneous thoughts in monte carlo tree search for code generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8103–8121, 2025.

[Lin *et al.*, 2022] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

[Liu *et al.*, 2025a] Jiaming Liu, Tao Cui, Tao Wang, Lishan Lin, Xi Zeng, Dazhi Lu, Shaoqing Jiao, Jun Wang, Xiaoyan Li, Shuyuan Xiao, et al. A multi-property optimizing generative adversarial network for de novo antimicrobial peptide design. *Advanced Science*, 12(38):e03443, 2025.

[Liu *et al.*, 2025b] Mengdi Liu, Xiaoxue Cheng, Zhangyang Gao, Hong Chang, Cheng Tan, Shiguang Shan, and

Xilin Chen. Protinvtree: Deliberate protein inverse folding with reward-guided tree search. *arXiv preprint arXiv:2506.00925*, 2025.

[Liu *et al.*, 2025c] Xuefeng Liu, Mingxuan Cao, Songhao Jiang, Xiao Luo, Xiaotian Duan, Mengdi Wang, Tobin R Sosnick, Jinbo Xu, and Rick Stevens. Monte carlo tree diffusion with multiple experts for protein design. *arXiv preprint arXiv:2509.15796*, 2025.

[Lu *et al.*, 2025] Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, and Meng Jiang. Learning to optimize multi-objective alignment through dynamic reward weighting. *arXiv preprint arXiv:2509.11452*, 2025.

[Meshchaninov *et al.*, 2024] Viacheslav Meshchaninov, Pavel Strashnov, Andrey Shevtsov, Fedor Nikolaev, Nikita Ivanisenko, Olga Kardymon, and Dmitry Vetrov. Diffusion on language model embeddings for protein sequence generation. *arXiv preprint arXiv:2403.03726*, 2024.

[Nanda *et al.*, 2017] Vikas Nanda, Sandeep V Belure, and Ofer M Shir. Searching for the pareto frontier in multi-objective protein design. *Biophysical reviews*, 9(4):339–344, 2017.

[Ngatchou *et al.*, 2005] Patrick Ngatchou, Anahita Zarei, and A El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th international conference on, intelligent systems application to power systems*, pages 84–91. IEEE, 2005.

[Pirtskhalava *et al.*, 2021] Malak Pirtskhalava, Anthony A Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, and Michael Tartakovsky. Dbaasp v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic acids research*, 49(D1):D288–D297, 2021.

[Plisson, 2022] Fabien Plisson. Overcoming the challenges in machine learning-guided antimicrobial peptide design. In *Proceedings of the 36th European and the 12th international peptide symposium, Sitges, Spain*, pages 207–210, 2022.

[Rathore *et al.*, 2024] Anand Singh Rathore, Shubham Choudhury, Akanksha Arora, Purva Tijare, and Gajendra PS Raghava. Toxinpred 3.0: An improved method for predicting the toxicity of peptides. *Computers in biology and medicine*, 179:108926, 2024.

[Repecka *et al.*, 2021] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

[Shi *et al.*, 2022] Guobang Shi, Xinyue Kang, Fanyi Dong, Yanchao Liu, Ning Zhu, Yuxuan Hu, Hanmei Xu, Xingzhen Lao, and Heng Zheng. Dramp 3.0: an enhanced comprehensive data repository of antimicrobial peptides. *Nucleic acids research*, 50(D1):D488–D496, 2022.

[Tang *et al.*, 2025] Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *ArXiv*, pages arXiv–2412, 2025.

[Tokuriki and Tawfik, 2009] Nobuhiko Tokuriki and Dan S Tawfik. Protein dynamism and evolvability. *Science*, 324(5924):203–207, 2009.

[Uehara *et al.*, 2025] Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and dna design. *arXiv preprint arXiv:2502.14944*, 2025.

[Veltri *et al.*, 2018] Daniel Veltri, Uday Kamath, and Amarda Shehu. Deep learning improves antimicrobial peptide recognition. *Bioinformatics*, 34(16):2740–2747, 2018.

[Wang *et al.*, 2024] Li Wang, Xiangzheng Fu, Jiahao Yang, Xinyi Zhang, Xiucai Ye, Yiping Liu, Tetsuya Sakurai, and Xiangxiang Zeng. Moformer: Multi-objective antimicrobial peptide generation based on conditional transformer joint multi-modal fusion descriptor. *arXiv preprint arXiv:2406.02610*, 2024.

[Wang *et al.*, 2025] Li Wang, Yiping Liu, Xiangzheng Fu, Xiucai Ye, Junfeng Shi, Gary G Yen, Quan Zou, Xiangxiang Zeng, and Dongsheng Cao. Hmamp: Designing highly potent antimicrobial peptides using a hypervolume-driven multiobjective deep generative model. *Journal of Medicinal Chemistry*, 68(8):8346–8360, 2025.

[Yao *et al.*, 2025] Lantian Yao, Jiahui Guan, Peilin Xie, Chia-Ru Chung, Zhihao Zhao, Danhong Dong, Yilin Guo, Wenyang Zhang, Junyang Deng, Yuxuan Pang, et al. dbamp 3.0: updated resource of antimicrobial activity and structural annotation of peptides in the post-pandemic era. *Nucleic acids research*, 53(D1):D364–D376, 2025.

[Yoon *et al.*, 2025] Jaesik Yoon, Hyeonseo Cho, Doojin Baek, Yoshua Bengio, and Sungjin Ahn. Monte carlo tree diffusion for system 2 planning. *arXiv preprint arXiv:2502.07202*, 2025.

[Yu *et al.*, 2023] Haoqing Yu, Ruheng Wang, Jianbo Qiao, and Leyi Wei. Multi-cgan: deep generative model-based multiproperty antimicrobial peptide design. *Journal of Chemical Information and Modeling*, 64(1):316–326, 2023.

[Zhao *et al.*, 2024] Dengwei Zhao, Shikui Tu, and Lei Xu. Efficient retrosynthetic planning with mcts exploration enhanced a* search. *Communications Chemistry*, 7(1):52, 2024.

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