EXPLORATORY BIOLOGICAL SEQUENCE DESIGN ON MASKED DISCRETE DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological sequence design under limited experimental budgets requires effective exploration of vast discrete sequence spaces under unreliable proxies trained from scarce data. Conservative search may remain concentrated around previously observed sequences, whereas broader exploration may reach regions where proxy predictions are uncertain. To reconcile conservative and broad exploration, we introduce a masked discrete diffusion framework **Reward-tilted Diverse Discrete Diffusion (RD3)**. Our method consists of three components to precisely control exploration. First, it constructs local neighborhoods of high-scoring sequences by partial masking and denoising. Second, we fine-tune the pretrained masked diffusion model to tilt sequence towards high-scoring regions during denoising. Finally, we apply categorical particle guidance, a variant of particle guidance in continuous space to promote diversity in the batch. Experimental results across nine benchmarks spanning DNA, RNA, peptide, and protein design demonstrate that RD3 effectively discovers high-fitness sequences, outperforming existing methods in terms of maximum fitness. Fitness–diversity and fitness–novelty analyses further demonstrate the effectiveness of RD3 in generating diverse and novel candidates without compromising fitness. Overall, the results highlight the value of combining search locality, reward alignment, and finite-sample diversification to balance fitness, novelty, and diversity under limited oracle budgets.

1 INTRODUCTION

Biological sequence design aims to discover novel sequences with desired functional properties in vast discrete sequence spaces, representing a challenge in protein engineering (Wu et al., 2021), gene regulation (Zrimec et al., 2022), and gene-delivery vector design (Bryant et al., 2021). Novel sequence discovery process involves iteratively proposing and evaluating batches of candidates, using evaluation feedback to guide subsequent proposals. Since evaluation cost of biological molecules is expensive (Angermueller et al., 2020; Jain et al., 2022), we need to discover desirable sequences within a limited evaluation budget. However, relying on exploitation of proxy models trained on limited data can concentrate proposals around a narrow set of high-scoring sequences, limiting exploration of other promising regions and potentially wasting evaluations on similar candidates.

In this paper, we propose Reward-tilted Diverse Discrete Diffusion (RD3), an exploration strategy for biological sequence design built on masked discrete diffusion models. RD3 comprises three techniques: **(i) partial masking and denoising** with masked diffusion model to explore controlled neighborhoods of given sequences, **(ii) fine-tuning the diffusion model** to sample from a reward-tilted prior distribution toward high-reward sequences, and **(iii) categorical particle guidance** adapted from Corso et al. (2024) to promote diversity within each sampling batch. Our novel contribution is twofolds. First, we use masked diffusion for local sequence exploration, reconstructing masked tokens using the full visible sequence context, while autoregressive reconstruction used by Angermueller et al. (2020); Kim et al. (2025a) relies only on preceding tokens. Second, we extend particle guidance from continuous diffusion to discrete token generation by reweighting categorical probabilities based on similarities among evolving candidates, discouraging similar completions during denoising.

We integrate RD3 into an active learning loop (Jain et al., 2022). We first pretrain a masked discrete diffusion model on the initial dataset and set it as a prior distribution. At each round, we update both the proxy model and the diffusion prior using the current measured dataset, and fine-tune

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the diffusion model using proxy-based rewards, tilting its sampling distribution toward high-utility regions. We then sample seed sequences from the current measured dataset, independently mask each token with probability δ, and reconstruct the masked tokens using particle-guided sampling to generate a candidate pool. A batch of candidates is selected for oracle evaluation, and the resulting measurements are added to the dataset for the next round.

2 Related Work

Discrete Diffusion Discrete diffusion models generate data over discrete spaces by learning to unmask a gradual corruption process (Austin et al., 2021; Campbell et al., 2022; Sun et al., 2023). Biological sequences fit the discrete setting because nucleotides and amino acids are drawn from finite alphabets, motivating applications of discrete diffusion to DNA and protein generation (Sarkar et al., 2024; Alamdari et al., 2023; Wang et al., 2024). Masked discrete diffusion (Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024) masks a sequence by replacing selected tokens with a mask symbol. Unlike left-to-right autoregressive models, masked discrete diffusion uses clean tokens to unmask over reverse steps without a fixed generation order, which preserves clean tokens during unmasking (Sahoo et al., 2024; Nie et al., 2025). Protein diffusion models uses partial-sequence generation by keeping clean residues or motifs fixed while generating the remaining sequence (Alamdari et al., 2023; Wang et al., 2024). RD3 uses a masked discrete diffusion prior to regenerate selected tokens of measured sequences while the other tokens remain unchanged.

Reward-Tilted and Diverse Diffusion Generation Pretrained diffusion models capture a data distribution, while design tasks require generation toward task-specific rewards. Inference-time methods modify sampling through guidance or sequential Monte Carlo without updating model parameters (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Nisonoff et al., 2025; Kim et al., 2025b). Training-based methods instead update the denoising model using reward feedback, including policy-gradient, direct reward optimization, and Soft-Q approaches (Black et al., 2024; Fan et al., 2023; Wang et al., 2025; Kang et al., 2026). Relative Trajectory Balance (RTB) trains a diffusion model to sample from a reward-reweighted distribution defined by the pretrained model and reward, while supporting off-policy training (Venkatraman et al., 2024). Reward alignment alone does not ensure diversity within a finite candidate batch. Particle Guidance (PG) couples sampling trajectories to promote diversity during generation (Corso et al., 2024). RD3 combines RTB-based reward alignment with categorical PG to promote both high utility and batch diversity.

Biological Sequence Optimization Model-guided biological sequence design uses several exploration strategies to discover high-fitness sequences under limited experimental evaluation. AdaLead (Sinai et al., 2020) and PEX (Ren et al., 2022) search locally by iteratively modifying promising measured sequences, whereas DbAS (Brookes & Listgarten, 2018) and CbAS (Brookes et al., 2019) update generative distributions toward sequences with desired properties. Model-based reinforcement learning optimizes a policy by a learned surrogate of the sequence landscape (Angermueller et al., 2020), while GFlowNet-based active learning (Jain et al., 2022) generates diverse candidate batches while accounting for epistemic uncertainty. To reduce reliance on unreliable proxy predictions for out-of-distribution sequences, δ-Conservative Search (Kim et al., 2025a) denoises partially masked high-scoring measured sequences and adjusts the search extent by proxy uncertainty. RD3 retains the conservative reconstruction principle but uses masked discrete diffusion as the sequence prior, then separately aligns generation with reward and promotes diversity within the candidate batch.

3 Preliminaries

3.1 Masked Discrete Diffusion

Masked discrete diffusion defines a generative process over discrete sequences through forward masking and reverse reconstruction (Sahoo et al., 2024).

Notation. Let $\mathcal{V}$ denote the vocabulary of biological tokens, and let $[\mathrm{MASK}]$ denote a masked token not contained in $\mathcal{V}$. We define $\bar{\mathcal{V}} = \mathcal{V} \cup \{[\mathrm{MASK}]\}$. A clean sequence is denoted by $x = (x_1, \ldots, x_L) \in \mathcal{V}^L$, where L is the sequence length and x_ℓ is the token at index $\ell \in \{1, \ldots, L\}$. For

diffusion time $t \in [0,1]$, let $x_t = (x_{t,1}, \ldots, x_{t,L}) \in \bar{\mathcal{V}}^L$ denote the noisy sequence at time t, with $x_0 = x$, where x_1 is a fully masked sequence. Here, $x_{t,\ell} \in \bar{\mathcal{V}}$ denotes the ℓ-th indexed token of sequence at diffusion time t. Let $m : [0,1] \to [0,1]$ denote the masking schedule, where $m(t)$ is the probability that a token is masked at diffusion time t. We assume $m(0) = 0$ and $m(1) = 1$.

Forward masking process. For a clean sequence $x \in \mathcal{V}^L$ and diffusion time t, let $q_t(x_t \mid x)$ denote the forward masking distribution over noisy sequences $x_t \in \bar{\mathcal{V}}^L$. With the indicator function $\mathbf{1}\{\cdot\}$, each position is masked independently according to

$$q_t(x_t \mid x) = \prod_{\ell=1}^{L} \left[(1 - m(t)) \mathbf{1}\{x_{t,\ell} = x_\ell\} + m(t) \mathbf{1}\{x_{t,\ell} = [\text{MASK}]\} \right]. \tag{1}$$

Reverse process. For $0 \le s < t \le 1$, let $p_\theta(x_s \mid x_t)$ denote the parameterized reverse transition from diffusion time t to an earlier time s, with the model parameters θ. Under the masked-diffusion reverse parameterization, masked tokens are progressively unmasked, while unmasked tokens remain unchanged (Sahoo et al., 2024). Repeated unmasking induces a distribution $p_\theta(x)$ over clean sequences, serving as the diffusion prior.

Training objective. Let $p_\theta(x_{0,\ell})$ denote the predicted clean token probability for $v \in \mathcal{V}$ at token index ℓ. We train the masked discrete diffusion model to unmask the original tokens at each masked indexes using cross-entropy loss:

$$\mathcal{L}_{\text{MDD}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D},\, t,\, x_t \sim q_t(\cdot|x)} \left[\sum_{\ell=1}^{L} \mathbf{1}\{x_{t,\ell} = [\text{MASK}]\} \log p_\theta(x_{0,\ell} = x_\ell \mid x_t) \right]. \tag{2}$$

3.2 Relative Trajectory Balance

Relative trajectory balance trains a posterior diffusion model to sample from a reward-tilted target distribution defined with respect to a fixed diffusion prior (Venkatraman et al., 2024).

Notation. Let $\delta \in [0,1]$ denote the masking probability. Starting from the diffusion time $t = \delta$, the prior p_θ runs reverse diffusion to $t = 0$, unmasking tokens while keeping the revealed tokens fixed.

Reward-tilted target distribution. Let $p_\theta(x)$ denote the prior distribution over clean sequences. Given a scalar reward function $r : \mathcal{V}^L \to \mathbb{R}$ and reward coefficient $\beta_R > 0$, the exact normalizing constant is $Z = \sum_{x \in \mathcal{V}^L} p_\theta(x) \exp(\beta_R r(x))$, where x ranges over all clean sequences. The reward-tilted target distribution is

$$p^*(x) = \frac{p_\theta(x) \exp(\beta_R r(x))}{Z}. \tag{3}$$

Trajectory balance. Let $\tau = (x_{t_K}, x_{t_{K-1}}, \ldots, x_{t_0})$ denote a reverse diffusion trajectory with $0 = t_0 < \cdots < t_K \le 1$, where K is the number of reverse diffusion steps and $x = x_{t_0}$ is the terminal clean sequence. Let $P_\theta(\tau)$ and $P_\psi(\tau)$ denote the probabilities assigned to τ by the prior and posterior reverse processes, respectively, where ψ denotes the posterior parameters. Relative trajectory balance imposes

$$Z_\omega \, P_\psi(\tau) = P_\theta(\tau) \exp(\beta_R r(x)), \tag{4}$$

where $Z_\omega > 0$ is a learned estimate of Z parameterized by ω. The corresponding loss is as follows:

$$\mathcal{L}_{\text{RTB}}(\tau; \psi, \omega) = \left[\log Z_\omega + \log \frac{P_\psi(\tau)}{P_\theta(\tau)} - \beta_R r(x) \right]^2. \tag{5}$$

When the balance relation holds for all reverse diffusion trajectories, the terminal distribution of the posterior equals $p^*(x)$ from Eq 3 (Venkatraman et al., 2024). During optimization, θ, r, and β_R are fixed, whereas ψ and ω are optimized. The loss can also be evaluated using trajectories sampled from a distribution other than the posterior being optimized, allowing off-policy training (Venkatraman et al., 2024).

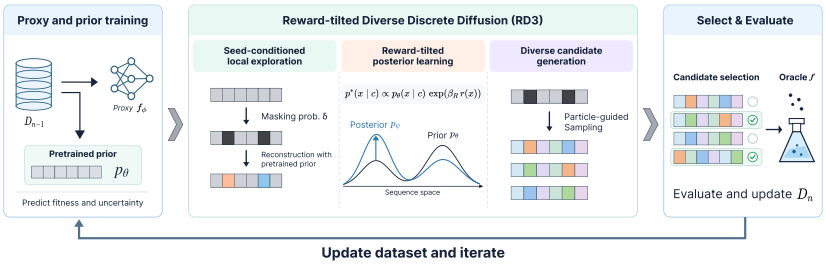


Figure 1: Active learning for biological sequence design with Reward-tilted Diverse Discrete Diffusion (RD3). At each round, the proxy model and diffusion prior are updated using the measured dataset. RD3 combines three techniques: (i) partial masking with probability δ and denoising with a masked diffusion model; (ii) fine-tuning to sample from a reward-tilted prior distribution; and (iii) categorical particle guidance. Selected candidates are evaluated by the oracle, and the resulting measurements are added to the dataset for the next round.

3.3 Particle Guidance

Particle Guidance (PG) (Corso et al., 2024) couples diffusion trajectories to promote diversity within a finite sample. Each denoising sample is called a particle.

Particle-guided diffusion. Let N_{PG} denote the number of particles and $x_t^{(i)}$ the continuous state of particle $i \in [1, \ldots, N_{\mathrm{PG}}]$ with time t increases along the denoising processes. Without PG, each particle denoises according to a sampling drift b_t and a diffusion coefficient g_t. Let $W_t^{(i)}$ denote independent Wiener processes supplying the noise. To introduce interactions between particles, PG uses a joint potential $\Phi_t(x_t^{(1)}, \ldots, x_t^{(N_{\mathrm{PG}})})$, which assigns a scalar value to the collection of particles. PG is formulated for continuous-state diffusion, and adds a guidance term to the sampling drift, yielding the following stochastic differential equation for each particle:

$$dx_t^{(i)} = \left[b_t(x_t^{(i)}) + g_t^2 \nabla_{x_t^{(i)}} \log \Phi_t \left(x_t^{(1)}, \ldots, x_t^{(N_{\mathrm{PG}})} \right) \right] dt + g_t \, dW_t^{(i)}. \tag{6}$$

4 Method

In this section, we introduce Reward-tilted Diverse Discrete Diffusion (RD3), which uses three exploration techniques within an active-learning framework: local search with masked discrete diffusion, reward alignment using Relative Trajectory Balance, and diversity sampling using categorical Particle Guidance. Figure 1 provides an overview of RD3.

4.1 Active Learning for Biological Sequence Design

We consider RD3 in an active-learning setting under a limited oracle budget. An expensive black-box oracle f assigns fitness $f(x)$ to a sequence x. The initial dataset $\mathcal{D}_0$ contains measured sequence–fitness pairs (x, y) with $y = f(x)$. Following prior active-learning approaches for biological sequence design (Angermueller et al., 2020; Jain et al., 2022; Kim et al., 2025a), we perform N rounds with a query budget of B sequences per round. At round $n \in \{1, \ldots, N\}$, $\mathcal{D}_{n-1}$ denotes the measured dataset available at the beginning of the round.

Step A. Proxy and Prior Training. At round n, we train a proxy ensemble f_ϕ on $\mathcal{D}_{n-1}$. Proxy training uses standardized fitness targets and an auxiliary classification objective, while only the regression predictions are used for acquisition. Let $\mu_n(x)$ and $\sigma_n(x)$ denote the predictive mean and standard deviation of the ensemble for sequence x. Given an uncertainty coefficient $\kappa \geq 0$, we use the upper confidence bound (UCB) (Srinivas et al., 2010).

$$a_n(x) = \mu_n(x) + \kappa \sigma_n(x). \tag{7}$$

The MDD prior p_θ is pretrained on $\mathcal{D}_0$. At round n, we continue prior training on $\mathcal{D}_{n-1}$ and denote the round-specific prior by p_{θ_n}. The prior p_{θ_n} is fixed for the remainder of round n. Proxy and prior training details are provided in Appendix C.

Step B. Candidate Generation with RD3. RD3 generates candidate sequences using the round-specific prior p_{θ_n} and acquisition score a_n. Candidate generation combines local exploration around measured sequences, reward-tilted posterior learning, and particle-guided sampling. Section 4.2 describes the three components.

Step C. Query Selection and Oracle Evaluation. Let $\mathcal{Q}_n$ denote the batch of B sequences selected from the generated candidates for oracle evaluation. After evaluating the selected sequences, the measured dataset is updated as

$$\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{(x, f(x)) : x \in \mathcal{Q}_n\}. \tag{8}$$

The query-selection procedure is described in Appendix C.7.

4.2 Reward-tilted Diverse Discrete Diffusion (RD3)

4.2.1 Local Exploration through Masked Discrete Diffusion

For the first exploration technique, we propose a local search that partially masks and unmasks measured sequences using a discrete diffusion model to explore neighborhoods of measured sequences. In active learning, proxy predictions can be unreliable outside the measured data distribution. To mitigate such distribution shifts, we partially mask measured sequences and reconstruct the masked positions with probability δ, preserving the unmasked tokens as context. Formally, given a seed sequence $x^{\text{seed}} \in \mathcal{D}_{n-1}$, we sample $\tilde{x} \sim q_\delta(\cdot \mid x^{\text{seed}})$.

We then reconstruct the masked positions using a masked discrete diffusion model. Although simple, using an autoregressive generator for unmasking presents a distinct disadvantage. Autoregressive models can only access preceding tokens, whereas biological sequences inherently contain dependencies at a global scale (Zhang et al., 2026). To mitigate this limitation, we introduce a masked diffusion model to unmask the sequences, allowing the model to attend to global sequence information simultaneously. So, starting from $\tilde{x}$, we complete the masked sequences through reverse transitions $p_\theta(x_s \mid x_t)$ to locally explore near the measured sequences.

4.2.2 Reward-Tilted Posterior Learning

To generate high-reward sequences from the partial-masked $\tilde{x}$, we use Relative Trajectory Balance (RTB) to learn a reward-tilted posterior from the MDD prior. The MDD prior defines a distribution over reconstructions of $\tilde{x}$, but its denoising objective does not explicitly shift probability mass toward high-reward sequences. We therefore train a posterior that tilts prior distribution to high reward-sequences.

We use the UCB utility $a(x)$ defined in equation 7 as the reward signal for posterior learning. Its uncertainty term retains an exploration incentive by assigning higher utility to candidates with uncertain proxy predictions. Let $r(x)$ denote the normalized UCB utility at round n. Given a partially masked sequence $\tilde{x}$, we define the reward-tilted target distribution as

$$p_n^*(x \mid \tilde{x}) \propto p_{\theta_n}(x \mid \tilde{x}) \exp(\beta_R a(x)), \tag{9}$$

where p_θ is the MDD prior and $\beta_R > 0$ controls the strength of reward tilting. We initialize the posterior p_ψ from the MDD prior and optimize it using the RTB objective in equation 5, so that the posterior approximates $p_n^*(x \mid \tilde{x})$. Reward normalization, reward-scale selection, and replay details are provided in Appendix B.2.

4.2.3 Categorical Particle Guidance

Finally, we apply categorical particle guidance to explore diverse candidates within each batch of biological sequences, by discouraging similar sequences from choosing the same token at the same index. Particle Guidance (PG) (Corso et al., 2024) improves diversity in finite sample by coupling samples during denoising, but was originally formulated for continuous-state diffusion.

Let $q_0^{(i,\ell)}(v)$ denote the base probability of clean token v to be chosen by a sequence i at masked index ℓ, predicted by the posterior denoiser before applying PG. To adapt PG to the discrete domain, we represent each masked token by $q_0^{(i,\ell)}(v)$, and each unmasked token by a one-hot vector with a single one at the observed token and zeros otherwise.

Let $x_t^{(i)}$ denote the sequence i at diffusion time t, and let $x_{t,\ell}^{(i)} \in \bar{\mathcal{V}}$ denote its token at index ℓ. For each sequence i and token index ℓ, define $\widehat{q}^{(i,\ell)}(v)$ the probability to choose clean token v as:

$$\widehat{q}^{(i,\ell)}(v) = \begin{cases} q_0^{(i,\ell)}(v), & x_{t,\ell}^{(i)} = [\text{MASK}], \\ \mathbf{1}[v = x_{t,\ell}^{(i)}], & x_{t,\ell}^{(i)} \neq [\text{MASK}]. \end{cases} \tag{10}$$

For each masked token index ℓ in sequence i, we measure similar sequences by averaging the dot products of their token probability vectors at corresponding indices, excluding ℓ; a higher similarity indicates a closer neighbor. The penalty $C_{i,\ell}(v)$ is then larger when sequences similar to sequence i are more likely to select v at the index ℓ.

With a PG coefficient $\beta_{\text{PG}} \geq 0$ and time scaling factor $(1-t)$, define the PG-adjusted probability of selecting clean token v as:

$$q_{\beta_{\text{PG}}}^{(i,\ell)}(v) \propto q_0^{(i,\ell)}(v) \exp[-\beta_{\text{PG}}(1-t)C_{i,\ell}(v)], \tag{11}$$

with normalization over $v \in \mathcal{V}$. Particle Guidance does not change the RTB objective.

Target-KL calibration A fixed β_{PG} applies one PG strength at every diffusion step and across every task, even though the needed strength varies with the timestep and the task. We therefore calibrate β_{PG} so that the PG effect, measured as a KL divergence, matches a target value t_{KL} chosen by the user. At each reverse diffusion step and at each masked token, we compute the KL divergence between the PG-adjusted token distribution $q_{\beta_{\text{PG}}}^{(i,\ell)}(v)$ and the base token distribution $q_0^{(i,\ell)}(v)$. We average the KL divergence over all masked tokens in the batch, then adjust β_{PG}, one shared value for the batch, until the averaged KL divergence approximately equals t_{KL}. See Appendix B.3.2.

5 Experiments

We evaluate RD3 under a common experimental setup across all experiments. Section 5.2 evaluates its ability to discover high-fitness sequences under a fixed oracle budget, and Section 5.3 examines the diversity and novelty of the discovered candidates. Section 5.4 assesses the contribution of each component of RD3, while Section 5.5 analyzes how key hyperparameters affect the resulting performance and generated sequences. We first describe the setup used throughout these evaluations.

5.1 Experimental Setup

Tasks and oracles. We evaluate RD3 on 9 tasks spanning DNA, RNA, peptide and protein design. Task details are provided in Appendix D.1. TF-Bind-8, the RNA landscapes, AAV, and GFP use the oracles provided in FLEXS (Sinai et al., 2020). Pab1 and E4B use the TAPE-based oracles (Rao et al., 2019) released by Kmicikiewicz et al. (2025). AMP uses the pretrained classifier of Jain et al. (2022). Every method starts from the same initial dataset and is scored by the same oracle.

Active learning setup. All methods run for 10 active learning rounds with a query budget of 128 sequences per round, except for AMP, which uses a budget of 1,024 sequences per round. We use a proxy that comprises an ensemble of three one-dimensional convolutional networks, following the architecture shared with the baselines (Sinai et al., 2020). See Appendix C.

Baselines. We compare RD3 against 12 established methods for iterative biological sequence design, covering evolutionary search (Sinai et al., 2020; Ren et al., 2022; Hansen & Ostermeier, 2001), reinforcement learning (Angermueller et al., 2020), GFlowNets (Jain et al., 2022; Kim et al., 2025a), Bayesian optimization (Snoek et al., 2012; Eriksson et al., 2019), generative model-based search (Brookes & Listgarten, 2018; Brookes et al., 2019; Kmicikiewicz et al., 2025) and directed evolution (Tran & Hy, 2025). Baseline configurations are provided in Appendix D.2.

Evaluation metrics. We evaluate the top 128 sequences selected by oracle score from the initial dataset and all queried sequences (Kim et al., 2025a). We report maximum, mean, and median fitness. Diversity is the average pairwise distance among the selected sequences, and novelty is the average distance from each selected sequence to its nearest sequence in the initial dataset (Jain et al., 2022). Results are reported as the mean and standard deviation over five independent runs. See Appendix D.3.

5.2 Sequence Fitness Optimization

Maximum Fitness Results Table 1 reports the maximum fitness achieved after 10 active learning rounds. RD3 achieves the highest maximum fitness on 7 out of 9 tasks and ranks second on Pab1. Figure 2 reports fitness progression over 10 rounds, where RD3 matches or exceeds the final fitness of the best baseline by round 6 on 5 out of 9 tasks. Additional results are provided in Appendix E.1.

Table 1: Maximum fitness of top 128 sequences obtained by each method. Reported by the mean and standard deviation over 5 runs. **Bold**: highest maximum fitness per task. Underline: second-highest.

Method	AAV	AMP	E4B	GFP	Pab1	RNA-A	RNA-B	RNA-C	TF-Bind-8
CMA-ES	0.500 ± 0.000	0.932 ± 0.000	7.743 ± 0.000	3.572 ± 0.000	0.843 ± 0.000	0.874 ± 0.100	0.786 ± 0.181	0.862 ± 0.116	0.993 ± 0.004
BO	0.512 ± 0.020	0.937 ± 0.005	7.743 ± 0.000	3.575 ± 0.007	0.843 ± 0.000	1.019 ± 0.026	1.009 ± 0.010	1.024 ± 0.082	0.993 ± 0.007
TuRBO	0.500 ± 0.000	0.932 ± 0.001	7.743 ± 0.000	3.572 ± 0.000	0.843 ± 0.000	1.053 ± 0.002	1.014 ± 0.001	1.094 ± 0.049	0.995 ± 0.006
DynaPPO	0.500 ± 0.000	0.932 ± 0.000	7.743 ± 0.000	3.572 ± 0.000	0.843 ± 0.000	0.777 ± 0.024	0.831 ± 0.027	0.840 ± 0.073	0.985 ± 0.002
PEX	0.659 ± 0.024	0.962 ± 0.001	7.887 ± 0.047	3.605 ± 0.000	$\mathbf{1.677 \pm 0.239}$	0.862 ± 0.039	0.722 ± 0.001	0.879 ± 0.015	0.994 ± 0.000
AdaLead	0.716 ± 0.019	0.946 ± 0.001	8.055 ± 0.027	3.588 ± 0.004	1.660 ± 0.296	0.981 ± 0.011	1.002 ± 0.013	0.983 ± 0.079	0.998 ± 0.003
MLDE	0.553 ± 0.004	–	8.009 ± 0.027	3.591 ± 0.003	0.946 ± 0.010	–	–	–	–
CbAS	0.522 ± 0.023	–	7.743 ± 0.000	3.584 ± 0.002	0.843 ± 0.000	0.771 ± 0.037	0.768 ± 0.036	0.805 ± 0.060	0.992 ± 0.007
DbAS	0.522 ± 0.023	–	7.743 ± 0.000	3.584 ± 0.002	0.843 ± 0.000	0.741 ± 0.033	0.744 ± 0.024	0.840 ± 0.067	0.993 ± 0.002
GFN-AL	0.500 ± 0.000	0.932 ± 0.000	7.743 ± 0.000	3.572 ± 0.000	0.843 ± 0.000	0.702 ± 0.037	0.696 ± 0.022	0.748 ± 0.043	0.977 ± 0.015
GFN-AL-δCS	0.708 ± 0.012	0.950 ± 0.008	8.051 ± 0.061	3.589 ± 0.002	1.600 ± 0.292	$\mathbf{1.055 \pm 0.000}$	1.014 ± 0.001	1.094 ± 0.050	0.982 ± 0.003
ProSpero	0.704 ± 0.023	–	8.090 ± 0.024	$\mathbf{3.615 \pm 0.002}$	1.408 ± 0.350	–	–	–	–
RD3	$\mathbf{0.879 \pm 0.023}$	$\mathbf{0.967 \pm 0.001}$	$\mathbf{8.136 \pm 0.023}$	3.596 ± 0.006	1.670 ± 0.380	$\mathbf{1.055 \pm 0.000}$	$\mathbf{1.015 \pm 0.000}$	$\mathbf{1.115 \pm 0.002}$	$\mathbf{0.999 \pm 0.003}$

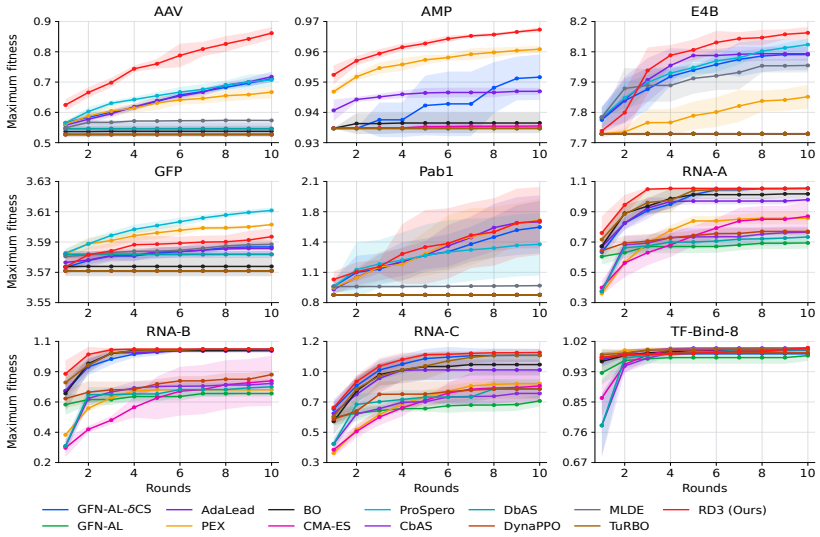


Figure 2: Maximum fitness over 10 active learning rounds across 9 biological sequence design tasks. Solid lines represent the mean across 5 runs, with shaded regions for the standard deviation. RD3 (Ours) is shown in red.

5.3 Fitness–Diversity and Novelty Trade-offs

Fitness–Diversity trade-off. Improving sequence fitness often trades off with diversity. Figure 3 compares the mean fitness and diversity of the final top-128 sequences across all methods. The masked discrete diffusion (MDD) baseline (gray star in the figure) uses MDD prior trained on measured data without reward-tilted fine-tuning or particle guidance. We use unnormalized Hamming distance for all tasks except TF-Bind-8 and AMP, which use unnormalized Levenshtein distance. On 7 out of

9 tasks, RD3 achieves both higher mean fitness and diversity than baseline, with the highest mean fitness and diversity on AAV, and the highest diversity among all methods on GFP.

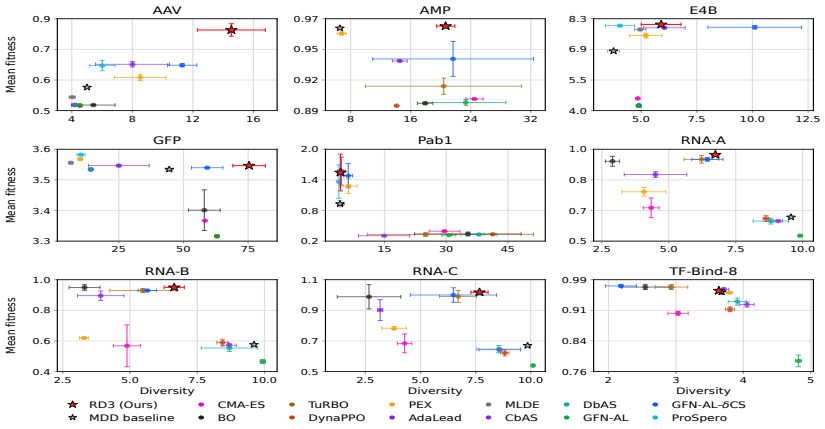


Figure 3: Mean fitness versus diversity of the retained set after 10 active learning rounds. Each point shows the mean over 5 runs, with horizontal and vertical bars for the standard deviation in diversity and in mean fitness, respectively.

Fitness–Novelty trade-off. Novelty measures the distance from the initial dataset rather than the spread among the retained sequences. Figure 8 in Appendix E.2 shows a similar trade-off between mean fitness and novelty. On 7 out of 9 tasks, no baseline achieves both higher mean fitness and higher novelty than RD3. RD3 achieves the highest novelty on AAV, GFP, and RNA-A, and the second-highest novelty on RNA-B and RNA-C.

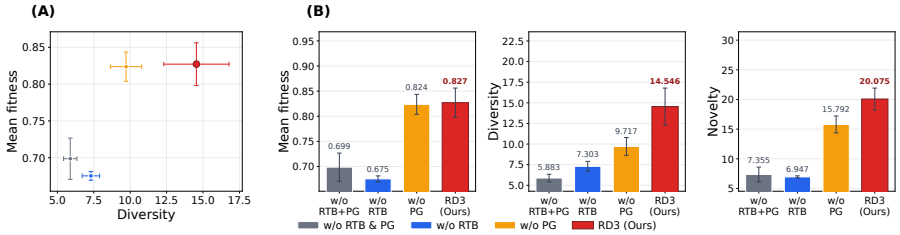


Figure 4: Component ablation on AAV. (A) Mean fitness versus diversity of the retained set across the four configurations. (B) The same configurations reported separately for mean fitness, diversity, and novelty. Points and bars show the mean over 5 runs, with error bars giving the standard deviation.

5.4 Ablation Study

To assess the contribution of individual components of RD3, the full method is compared against three ablated variants: (i) without reward-tilted fine-tuning, (ii) without particle guidance, and (iii) without both. Figure 4 shows the results on AAV. See Appendix D.4 for further experimental details.

Ablation on reward-tilted fine-tuning. We examine the effect of reward-tilted fine-tuning by sampling from the prior instead of the fine-tuned posterior. As shown in Figure 4, removing reward-tilted fine-tuning reduces performance across all metrics on AAV, with mean fitness and novelty decreased on RNA-C and E4B in Appendix E.3. These results indicate that reward-tilted fine-tuning improves both fitness optimization and novelty.

Ablation on particle guidance. We analyze the effect of PG by sampling candidate batches without PG while keeping the trained posterior fixed. Shown in Figure 4, diversity and novelty decrease on AAV, while mean fitness is unchanged, with a similar pattern on RNA-C and E4B (Appendix E.3). PG therefore promotes diversity and novelty during sampling without reducing mean fitness.

5.5 Sensitivity Analysis

This section examines RD3's sensitivity on AAV to the target KL divergence t_{KL}, the masking probability δ, and the reward coefficient β_R, all of which decide how far generation moves from the measured data. We vary each of them in turn while holding the other two at the headline configuration.

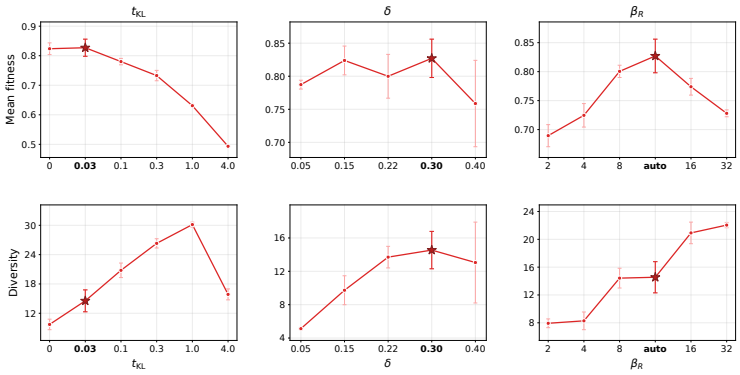


Figure 5: Sensitivity analysis on AAV. Each column varies one hyperparameter while keeping the others fixed. Stars indicate the RD3 configuration. The automatically selected reward strength (β_R) is 8.47 ± 0.29 across runs. Results are reported as mean $\pm$ standard deviation over 3 runs. Appendice D.4 and E.4 provide further details and the underlying values for all 5 metrics.

Analysis on the target KL divergence t_{KL}. As shown in Figure 5, the setting used by RD3 reaches the highest mean fitness across the range, and the largest t_{KL} loses diversity as well as fitness.

Analysis on the masking probability δ. The masking probability sets how many positions of the seed are reopened for reconstruction. A small value keeps candidates close to the sequences already scored, while a large value weakens the anchor to the seed. Figure 5 shows that diversity increases up to the setting used by RD3, while the largest value degrades both fitness and diversity.

Analysis on the reward coefficient β_R. A lower β_R produces a weaker reward tilt, keeping the posterior closer to the prior, whereas a higher β_R places greater weight on the proxy reward and may move candidates toward regions where proxy predictions are less reliable. As shown in Figure 5, mean fitness rises to a single peak across the range, while diversity increases throughout. RD3 sets β_R once per run using the entropy of the pretrained prior.

6 Conclusion

In this paper, we introduced Reward-tilted Diverse Discrete Diffusion (RD3), a masked discrete diffusion framework for biological sequence design under a limited oracle budget. Partial masking of measured sequences holds generation near the data the proxy was fitted on, and relative trajectory balance tilts the prior toward high acquisition values while anchoring the posterior to the prior. Particle guidance repels the candidates drawn within one batch and raises diversity at matched fitness. RD3 improves over the best record of the initial dataset on every task and attains the highest maximum fitness on 7 of 9 benchmark instances spanning DNA, RNA, peptide, and protein design.

Limitations and future works. RD3 starts from observed sequences and restricts edits through partial masking, which favors local exploration and may limit the discovery of high-fitness sequences far from the observed data. Posterior training and candidate selection rely on a learned proxy, whose prediction errors may misdirect the search. Benchmark oracles stand in for laboratory measurement, so improvements in benchmark fitness do not establish improvements in measured function. Validating selected sequences in wet-lab assays and extending the reward to multiple properties, such as binding affinity and stability, remain for future work.

AI USE STATEMENT

Large language models (LLMs) were used for minor language editing to improve grammar, clarity, and readability, and for limited code assistance involving syntax debugging and function refactoring. LLMs did not contribute to the conception of the research problem, development of the core methodology, or design and execution of experiments. All research ideas, methodological choices, analyses, and interpretations of results originated with the authors. The authors reviewed and verified the final manuscript and take full responsibility for its content.

REFERENCES

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Lu, Nicolo Fusi, Ava Amini, and Kevin Yang. Protein generation with evolutionary diffusion. In *NeurIPS 2023 Workshops: GenBio*, 2023.

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*, 2020.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

Luis A. Barrera, Anastasia Vedenko, Jesse V. Kurland, Julia M. Rogers, Stephen S. Gisselbrecht, Elizabeth J. Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, Trevor Siggers, Leila Shokri, Raluca Gordân, Nidhi Sahni, Chris Cotsapas, Tong Hao, Song Yi, Manolis Kellis, Mark J. Daly, Marc Vidal, David E. Hill, and Martha L. Bulyk. Survey of variation in human transcription factors reveals prevalent DNA binding changes. *Science*, 351(6280):1450–1454, 2016. doi: 10.1126/science.aad2257.

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

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 773–782. PMLR, 2019.

David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

Drew H Bryant, Ali Bashir, Sam Sinai, Nina K Jain, Pierce J Ogden, Patrick F Riley, George M Church, Lucy J Colwell, and Eric D Kelsic. Deep diversification of an aav capsid protein by machine learning. *Nature biotechnology*, 39(6):691–696, 2021.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://openreview.net/forum?id=DmT862YAieY.

Gabriele Corso, Yilun Xu, Valentin de Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: Non-I.I.D. diverse sampling with diffusion models. In *International Conference on Learning Representations*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3497.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. doi: 10.48550/arXiv.2207.12598.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9786–9801. PMLR, 2022.

Hyeongyu Kang, Jaewoo Lee, Woocheol Shin, Kiyoung Om, and Jinkyoo Park. Diffusion fine-tuning via reparameterized policy gradient of the soft q-function. In *International Conference on Learning Representations*, 2026.

Hyeonah Kim, Minsu Kim, Taeyoung Yun, Sanghyeok Choi, Emmanuel Bengio, Alex Hernández-García, and Jinkyoo Park. Improved off-policy reinforcement learning in biological sequence design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30290–30315. PMLR, 2025a.

Minsu Kim, Federico Berto, Sungsoo Ahn, and Jinkyoo Park. Bootstrapped training of score-conditioned generator for offline design of biological sequences. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67643–67661, 2023.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d9042abf40782fbce28901c1c9c0e8d8-Abstract-Conference.html.

Michal Kmicikiewicz, Vincent Fortuin, and Ewa Szczurek. ProSpero: Active learning for robust protein design beyond wild-type neighborhoods. In *Advances in Neural Information Processing Systems*, volume 38, pp. 145223–145256, 2025. doi: 10.52202/085713-4364.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011. doi: 10.1186/1748-7188-6-26.

Daniel Melamed, David L. Young, Caitlin E. Gamble, Christina R. Miller, and Stanley Fields. Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly(a)-binding protein. *RNA*, 19(11):1537–1551, 2013. doi: 10.1261/rna.040709.113.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations*, 2025.

Pierce J. Ogden, Eric D. Kelsic, Sam Sinai, and George M. Church. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science*, 366(6469): 1139–1143, 2019. doi: 10.1126/science.aaw2900.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Zhizhou Ren, Jiahan Li, Fan Ding, Yuan Zhou, Jianzhu Ma, and Jian Peng. Proximal exploration for model-guided protein sequence design. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18520–18536. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ren22a.html.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4135.

Anirban Sarkar, Ziqi Tang, Chris Z Zhao, and Peter K Koo. Designing DNA with tunable regulatory activity using discrete diffusion. In *NeurIPS 2024 Workshops: AIDrugX*, 2024.

Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva, Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016. doi: 10.1038/nature17995.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3277. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D Kelsic. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv preprint arXiv:2010.02141*, 2020.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1015–1022, 2010.

Lea M. Starita, Jonathan N. Pruneda, Russell S. Lo, Douglas M. Fowler, Helen J. Kim, Joseph B. Hiatt, Jay Shendure, Peter S. Brzovic, Stanley Fields, and Rachel E. Klevit. Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14):E1263–E1272, 2013. doi: 10.1073/pnas.1303309110.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=BYWWwSY2G5s.

Thanh V. T. Tran and Truong Son Hy. Protein design by directed evolution guided by large language models. *IEEE Transactions on Evolutionary Computation*, 29(2):418–428, 2025. doi: 10.1109/TEVC.2024.3439690.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, Alexandre Adam, Jarrid Rector-Brooks, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Amortizing intractable inference in diffusion

models for vision, language, and control. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2422.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen Wang, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *International Conference on Learning Representations*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52309–52333. PMLR, 2024.

Zachary Wu, Kadina E Johnston, Frances H Arnold, and Kevin K Yang. Protein sequence design with deep generative models. *Current opinion in chemical biology*, 65:18–27, 2021.

Xiang Zhang, Jiaqi Wei, Zijie Qiu, Sheng Xu, Zhi Jin, ZhiQiang Gao, Nanqing Dong, and Siqi Sun. Bidirectional representations augmented autoregressive biological sequence generation. *Advances in Neural Information Processing Systems*, 38:51645–51672, 2026.

Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K Speicher, Christoph S Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, et al. Controlling gene expression with deep generative design of regulatory dna. *Nature communications*, 13(1):5099, 2022.

A ALGORITHM

Algorithm 1 Active Learning with RD3

Input: oracle f, initial dataset $\mathcal{D}_0$, rounds N, query budget B, pool size C

Pretrain $p_{\theta_{\text{pre}}}$ on $\mathcal{D}_0$; $\beta_R \leftarrow S_{\text{pre}}/z_0^{\max}$
for $n = 1$ **to** N **do**
  ▷ STEP A: PROXY AND PRIOR TRAINING
    Train $\{f_{\phi_j}\}_{j=1}^{J}$ on $\mathcal{D}_{n-1}$; $a \leftarrow \mu + \kappa\,\sigma$ ▷ Eq. 28, 7
    $p_{\theta_n} \leftarrow$ prior refit on $\mathcal{D}_{n-1}$, then frozen
  ▷ STEP B: CANDIDATE GENERATION WITH RD3
    $p_{\psi_n} \leftarrow$ posterior trained on $\mathcal{D}_{n-1}$ ▷ Algorithm 3
    $\mathcal{X} \leftarrow$ candidate pool of size C ▷ Algorithm 4
  ▷ STEP C: QUERY SELECTION AND ORACLE EVALUATION
    $\mathcal{Q}_n \leftarrow B$ sequences selected from $\mathcal{X}$ by a and distance
    $\mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(x, f(x)) : x \in \mathcal{Q}_n\}$ ▷ Eq. 8
end for

Algorithm 2 Conservative Masking

Input: dataset $\mathcal{D}$, number of seeds M, masking probability δ

$x_1^{\text{seed}}, \ldots, x_M^{\text{seed}} \sim \mathcal{D}$ ▷ rank-based weights, Eq. 12
$\mathcal{M}^{(i)} \leftarrow \{\ell : b_\ell^{(i)} = 1\}$, $b_\ell^{(i)} \sim \text{Bernoulli}(\delta)$, at least one per seed
$\tilde{x}_\ell^{(i)} \leftarrow$ [MASK] if $\ell \in \mathcal{M}^{(i)}$, else $x_\ell^{\text{seed},(i)}$
return $\tilde{x}_1, \ldots, \tilde{x}_M$, $t_K = \delta$

Algorithm 3 Reward-Tilted Fine-Tuning

Input: prior p_θ, dataset $\mathcal{D}$, acquisition a, reward coefficient β_R, masking probability δ, steps E, batch size M, replay ratio ρ

$\psi \leftarrow \theta$; initialize $\log Z_\omega$; $\mathcal{R} \leftarrow \emptyset$; $\bar{a}, s_a \leftarrow$ mean and sd of $\{a(x) : (x, y) \in \mathcal{D}\}$
for $e = 1$ **to** E **do**
  $\tilde{x}_1, \ldots, \tilde{x}_M, t_K \leftarrow$ masked seeds from $\mathcal{D}$ ▷ Algorithm 2
  $\tau_i \sim \tilde{P}_\psi(\cdot \mid \tilde{x}_i)$; $\hat{x}_i \leftarrow x_{t_0}$ of τ_i ▷ behavior policy, Appendix C
  $r_i \leftarrow \text{clip}\big((a(\hat{x}_i) - \bar{a})/s_a,\, -50,\, 50\big)$ ▷ Eq. 15
  $\Delta_i \leftarrow \log Z_\omega(\tilde{x}_i) + \log \dfrac{P_\psi(\tau_i \mid \tilde{x}_i)}{P_\theta(\tau_i \mid \tilde{x}_i)} - \beta_R\, r_i$ ▷ Eq. 20
  $\Delta_r \leftarrow$ residuals of $\lfloor \rho M \rfloor$ trajectories drawn from $\mathcal{R}$, once $|\mathcal{R}| \geq M$
  Update ψ, ω on the mean of $\{\Delta_i^2\}$ and $\{\Delta_r^2\}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{\tau_i\}$
end for
return p_ψ

Algorithm 4 Categorical Particle Guidance

Input: posterior p_ψ, dataset $\mathcal{D}$, pool size C, particles N_{PG}, reverse steps K, target KL t_{KL}

$\mathcal{X} \leftarrow \emptyset$
for $\lceil C/N_{\mathrm{PG}} \rceil$ batches **do**
  $x^{(1)}, \ldots, x^{(N_{\mathrm{PG}})}, t_K \leftarrow$ masked seeds from $\mathcal{D}$ ▷ Algorithm 2
  Reverse-time grid $t_K > t_{K-1} > \cdots > t_0 = 0$
  for $k = K$ **down to** 1 **do**
    $q_0^{(i,\ell)} \leftarrow p_\psi(\cdot \mid x_{t_k}^{(i)}, t_k)$ ▷ clean-token distribution
    $C_{i,\ell} \leftarrow$ repulsion cost from $\{\hat{q}^{(j,\ell)}\}_{j \neq i}$ ▷ Eq. 24
    $q_\beta^{(i,\ell)} \propto q_0^{(i,\ell)} \exp\big[-\beta(1-t_k)\, C_{i,\ell}\big]$
    $\beta_{\mathrm{PG}} \leftarrow$ the value with $F_{t_k}(\beta_{\mathrm{PG}}) = t_{\mathrm{KL}}$
    Reveal each masked position of $x^{(i)}$ with probability $1 - t_{k-1}/t_k$ by sampling $q_{\beta_{\mathrm{PG}}}^{(i,\ell)}$
  end for
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{x^{(1)}, \ldots, x^{(N_{\mathrm{PG}})}\}$
end for
return the first C entries of $\mathcal{X}$

B Additional Methodological Details

B.1 Conservative Masked Diffusion

Rank-based seed sampling. At active-learning round n, a seed sequence is sampled from $\mathcal{D}_{n-1}$. For $(x^{(i)}, y_i) \in \mathcal{D}_{n-1}$, let $\mathrm{rank}(x^{(i)})$ denote the rank of y_i among the fitness values in $\mathcal{D}_{n-1}$, with rank 1 assigned to the highest fitness. Let $k > 0$ denote the rank-weighting coefficient. Following rank-based prioritization (Kim et al., 2025a), the sampling weight of $x^{(i)}$ is

$$w_i \propto \frac{1}{k|\mathcal{D}_{n-1}| + \mathrm{rank}(x^{(i)})}. \tag{12}$$

The weights are normalized before sampling, and the sampled sequence is denoted by x^{seed}.

Partial masking. Let $x^{\mathrm{seed}} = (x_1^{\mathrm{seed}}, \ldots, x_L^{\mathrm{seed}})$ be a fixed-length seed sequence. Let $\mathcal{M} \subseteq \{1, \ldots, L\}$ denote the set of masked positions, where each position is included independently with probability δ. If $\mathcal{M}$ is empty, one position is sampled uniformly from $\{1, \ldots, L\}$ and added to $\mathcal{M}$. The partially masked sequence $\tilde{x}$ is

$$\tilde{x}_\ell = \begin{cases} [\mathrm{MASK}], & \ell \in \mathcal{M}, \\ x_\ell^{\mathrm{seed}}, & \ell \notin \mathcal{M}. \end{cases} \tag{13}$$

Under the linear masking schedule $m(t) = t$, reverse diffusion starts from $\tilde{x}$ at $t = \delta$. The nonempty-mask correction gives $\mathbb{E}[|\mathcal{M}|] = \delta L + (1 - \delta)^L$.

Let $d_H(x, x')$ denote the Hamming distance between two equal-length sequences. Under the SUBS parameterization, positions outside $\mathcal{M}$ remain unchanged during reverse diffusion (Sahoo et al., 2024). Therefore, a reconstructed sequence x satisfies

$$d_H(x, x^{\mathrm{seed}}) \leq |\mathcal{M}|. \tag{14}$$

A masked position may be reconstructed to its original token, so $|\mathcal{M}|$ is an upper bound on the number of changed positions.

Variable-length masking for AMP. For AMP, let L_{seed} denote the length of x^{seed}, and let p_{len} denote the categorical distribution over sequence lengths used at round n. A target length is sampled as $L_{\mathrm{target}} \sim p_{\mathrm{len}}$. Let $[\mathrm{PAD}]$ denote the padding token, with $[\mathrm{PAD}] \notin \mathcal{V}$. Positions $1, \ldots, L_{\mathrm{target}}$ are called active positions.

Bernoulli masking with probability δ is first applied to the padded seed representation. If $L_{\mathrm{target}} > L_{\mathrm{seed}}$, positions $L_{\mathrm{seed}} + 1, \ldots, L_{\mathrm{target}}$ are initialized as $[\mathrm{MASK}]$. Positions after L_{target} are then fixed to $[\mathrm{PAD}]$. The sampled target length remains fixed throughout reverse diffusion.

The nonempty-mask correction is applied before positions after L_{target} are replaced by $[\mathrm{PAD}]$. Therefore, when $L_{\mathrm{target}} \leq L_{\mathrm{seed}}$, the correction does not guarantee that a masked position remains in the active region. If no active position is masked after imposing the target length, reverse diffusion performs no token update and returns the seed restricted to the first L_{target} positions. The construction of p_{len} and the padded representation are described in Appendix C.

B.2 Reward-Tilted Posterior Learning

B.2.1 Reward Normalization and Reward Scale

At active-learning round n, let $\bar{a}$ and s_a denote the mean and standard deviation of the acquisition scores $\{a(x) : (x, y) \in \mathcal{D}_{n-1}\}$. Let $\mathrm{clip}(u, l, h)$ denote truncation of u to the interval $[l, h]$. The reward used for RTB is

$$r(x) = \mathrm{clip}\left(\frac{a(x) - \bar{a}}{s_a}, -50, 50\right). \tag{15}$$

The values $\bar{a}$ and s_a are computed before RTB optimization and remain fixed during round n.

Let $\bar{y}_0$ and $s_{y,0}$ denote the mean and population standard deviation of the fitness values in $\mathcal{D}_0$, and let $y_0^{\max} = \max_{(x,y)\in\mathcal{D}_0} y$. Define the standardized maximum fitness as

$$z_0^{\max} = \frac{y_0^{\max} - \bar{y}_0}{s_{y,0}}. \tag{16}$$

Let $p_{\theta_{\text{pre}}}$ denote the MDD prior after pretraining on $\mathcal{D}_0$. For a categorical distribution p over $\mathcal{V}$, let $H(p) = -\sum_{v\in\mathcal{V}} p(v)\log p(v)$ denote its entropy. For a fixed-length sequence of length L, define the pretrained entropy scale as

$$S_{\text{pre}} = \sum_{\ell=1}^{L} H\big(p_{\theta_{\text{pre}}}\left(x_{0,\ell} = \cdot \mid x_1 = [\text{MASK}]^L\right)\big)\,. \tag{17}$$

The reward coefficient is

$$\beta_R = \frac{S_{\text{pre}}}{z_0^{\max}}. \tag{18}$$

The coefficient β_R is computed once before active learning and remains fixed across rounds.

For AMP, let $p_{\text{len},0}$ denote the length distribution estimated from $\mathcal{D}_0$. The pretrained entropy scale includes the entropy of the length distribution:

$$S_{\text{pre}} = H(p_{\text{len},0}) + \mathbb{E}_{L\sim p_{\text{len},0}}\left[\sum_{\ell=1}^{L} H\big(p_{\theta_{\text{pre}}}\left(x_{0,\ell} = \cdot \mid x_1 = [\text{MASK}]^L\right)\big)\right]. \tag{19}$$

Only amino-acid positions contribute to the token-entropy term. The numerical evaluation of equation 19 is described in Appendix C.

B.2.2 Conditional RTB Objective

Let p_ψ denote the posterior diffusion model trained with RTB. For a partially masked sequence $\tilde{x}$, let τ denote a reverse trajectory from $\tilde{x}$ to a clean sequence x. Let $P_\theta(\tau \mid \tilde{x})$ and $P_\psi(\tau \mid \tilde{x})$ denote the trajectory probabilities under the prior p_θ and posterior p_ψ, respectively. Let ω denote the parameters of a positive normalizing model $Z_\omega(\tilde{x})$. The conditional RTB relation is

$$Z_\omega(\tilde{x})P_\psi(\tau \mid \tilde{x}) = P_\theta(\tau \mid \tilde{x})\exp(\beta_R r(x))\,. \tag{20}$$

Marginalizing equation 20 over reverse trajectories gives

$$p^*(x \mid \tilde{x}) \propto p_\theta(x \mid \tilde{x})\exp(\beta_R r(x))\,. \tag{21}$$

For AMP, L_{target} determines the active positions and remains fixed throughout the RTB trajectory. Positions fixed to [PAD] are excluded from reverse transitions. If no active position is masked, the trajectory contains no token update.

B.3 Categorical Particle Guidance

B.3.1 Sequence Soft Neighbor Repulsion

Let N_{PG} denote the number of particles in a sampling batch. For particle i and position ℓ, let $\widehat{q}^{(i,\ell)}$ denote the effective token distribution defined in Eq. (10). For two categorical distributions p and q over $\mathcal{V}$, define their inner product as $\langle p, q\rangle = \sum_{v\in\mathcal{V}} p(v)q(v)$.

For fixed-length sequences with $L > 1$, the similarity between particles i and j at position ℓ is

$$s_{ij,\ell} = \frac{1}{L-1}\sum_{h\neq\ell}\left\langle \widehat{q}^{(i,h)}, \widehat{q}^{(j,h)}\right\rangle, \tag{22}$$

where the sum over h ranges from 1 to L.

Let $\tau_s > 0$ denote the neighbor temperature. For $j \neq i$, define the neighbor weights by

$$w_{ij,\ell} \propto \exp\left(\frac{s_{ij,\ell}}{\tau_s}\right), \qquad \sum_{j\neq i} w_{ij,\ell} = 1. \tag{23}$$

The repulsion cost for token v at position ℓ is

$$C_{i,\ell}(v) = \sum_{j\neq i} w_{ij,\ell}\widehat{q}^{(j,\ell)}(v). \tag{24}$$

Variable-length interactions for AMP. For AMP, a position is active for a particle if the position does not exceed the particle's target length. Pairwise similarity excludes position ℓ and is computed over positions that are active in at least one of particles i and j. A position fixed to [PAD] in both particles is excluded from the similarity and its normalization. A position fixed to [PAD] in one particle and active in the other remains in the normalization and contributes zero token agreement. If no eligible position remains, $s_{ij,\ell}$ is set to zero.

If particle j is fixed to [PAD] at position ℓ, its contribution to $C_{i,\ell}(v)$ is zero for every $v \in \mathcal{V}$. Particle Guidance is applied only to masked active positions. A particle with no masked active position receives no guidance update.

B.3.2 TARGET-KL CALIBRATION

For categorical distributions q and p over $\mathcal{V}$, let $D_{\mathrm{KL}}(q\|p) = \sum_{v\in\mathcal{V}} q(v)\log\frac{q(v)}{p(v)}$ denote the Kullback–Leibler divergence. At diffusion time t, let $\mathcal{O}_t$ denote the set of masked particle–position pairs in the sampling batch. For AMP, pairs corresponding to positions fixed to [PAD] are excluded from $\mathcal{O}_t$.

For $\beta \geq 0$, let $q_\beta^{(i,\ell)}$ denote the categorical distribution obtained by replacing β_{PG} with β in equation 11. For $|\mathcal{O}_t| > 0$, define the mean local KL divergence as

$$F_t(\beta) = \frac{1}{|\mathcal{O}_t|} \sum_{(i,\ell)\in\mathcal{O}_t} D_{\mathrm{KL}}\left(q_\beta^{(i,\ell)} \| q_0^{(i,\ell)}\right). \tag{25}$$

Let $t_{\mathrm{KL}} \geq 0$ denote the target mean KL divergence. At each reverse diffusion step, one value of β_{PG} is shared across the sampling batch and selected such that $F_t(\beta_{\mathrm{PG}}) \approx t_{\mathrm{KL}}$. Setting $t_{\mathrm{KL}} = 0$ disables Particle Guidance. If $\mathcal{O}_t$ is empty, no calibration or guidance update is performed. The numerical calibration procedure is described in Appendix C.

C Implementation Details

C.1 Proxy Training

The proxy ensemble contains three independently trained networks. For RNA-A/B/C, AAV, Pab1, E4B, and GFP, each proxy uses a one-dimensional CNN with 32 filters, kernel size 5, and a 128-dimensional dense layer (Sinai et al., 2020). The FLEXS-based baselines and GFN-AL-δCS use the same proxy architecture and ensemble size (Sinai et al., 2020; Kim et al., 2025a). TF-Bind-8 and AMP use two-layer MLP proxies with hidden dimension 2,048 and dropout probability 0.1.

Regression targets are standardized before proxy training. Let $\bar{y}$ and $s_y > 0$ denote the mean and standard deviation used for normalization. For a measured fitness y, define

$$z(y) = \frac{y - \bar{y}}{s_y}. \tag{26}$$

Each proxy also contains an auxiliary binary classification head for high-fitness sequences. Let y_{80} denote the empirical 80th-percentile fitness threshold. For a measured fitness y, define the binary target

$$b(y) = \mathbf{1}\{y \geq y_{80}\}. \tag{27}$$

For a measured pair (x, y), let $\hat{z}(x)$ denote the standardized fitness prediction and let $\hat{b}(x) \in (0, 1)$ denote the predicted probability of the high-fitness class. Each ensemble member is trained using

$$\mathcal{L}_{\text{proxy}}(x, y) = \big(\hat{z}(x) - z(y)\big)^2 - b(y) \log \hat{b}(x) - \big(1 - b(y)\big) \log\big(1 - \hat{b}(x)\big). \tag{28}$$

Proxy training minimizes the minibatch average of Eq. equation 28. Only the regression output is used for acquisition. Before computing the ensemble mean and standard deviation in Eq. equation 7, the standardized regression prediction is mapped back to the original fitness scale as

$$\hat{y}(x) = \bar{y} + s_y \hat{z}(x). \tag{29}$$

For RNA-A/B/C, AAV, Pab1, E4B, and GFP, proxy parameters are warm-started across active-learning rounds, and $\bar{y}$, s_y, and y_{80} are fixed to their initial-round values. For TF-Bind-8 and AMP, proxy parameters are reinitialized at each round, and $\bar{y}$, s_y, and y_{80} are recomputed from $\mathcal{D}_{n-1}$.

Training minibatches use rank-prioritized sampling. Let N_{tr} denote the number of sequences in the training split, and let $\text{rank}(x)$ denote the rank of sequence x according to standardized fitness, with rank 1 assigned to the highest-fitness sequence. The sampling weight is proportional to

$$\frac{1}{0.01 N_{\text{tr}} + \text{rank}(x)}. \tag{30}$$

The coefficient 0.01 follows the proxy-training procedures of Sinai et al. (2020); Kim et al. (2025a).

The CNN proxies are trained with Adam using learning rate 10^{-3}, weight decay 10^{-4}, batch size 256, and dropout probability 0.25. Training runs for at most 3,000 epochs. Early stopping uses the held-out regression error with patience 10 and minimum improvement 10^{-5}. The validation split contains approximately 10% of the available data, with at least 32 validation sequences when the dataset size permits.

The MLP proxies are trained with Adam using learning rate 10^{-4}, weight decay 10^{-4}, and batch size 256. TF-Bind-8 and AMP are trained for at most 3,000 and 30,000 optimization steps, respectively. Validation is performed every 50 optimization steps using the held-out regression error, with patience 10 and minimum improvement 10^{-5}.

C.2 Masked Diffusion Prior

The diffusion prior used the x_0-prediction SUBS parameterization (Sahoo et al., 2024) with the linear masking schedule $m(t) = t$. During training, diffusion time was sampled uniformly from $[10^{-5}, 1]$. The loss in equation 2 was evaluated over masked positions without additional time-dependent weighting. For AMP, positions fixed to [PAD] were excluded from the loss.

Table 2: Task-specific parameters used by RD3.

Task	κ	δ	k	η	Prefilter	$K_{\max}$
TF-Bind-8	0.1	0.50	0.01	1.0	8,192	32
RNA-A/B/C	1.0	0.50	0.01	1.0	8,192	32
AMP	0.1	0.50	0.01	5.0	8,192	32
AAV	1.0	0.30	0.01	5.0	2,048	32
Pab1	1.0	0.1333	0.01	5.0	2,048	32
E4B	1.0	0.0980	0.01	5.0	2,048	32
GFP	1.0	0.30	0.001	5.0	2,048	16

The denoising network was a six-layer Transformer with model dimension 256, eight attention heads, and feed-forward dimension 1,024. Let K denote the number of reverse diffusion steps and $K_{\max}$ its task-specific upper bound. We used

$$K = \min(K_{\max}, L).$$

The values of $K_{\max}$ are reported in Table 2. Fully masked generation used a reverse-time grid from $t = 1$ to $t = 0$, whereas seed-conditioned generation started from $t = \delta$ and ended at $t = 0$.

The prior was pretrained on $\mathcal{D}_0$ for 1,500 epochs using AdamW with batch size 256, learning rate 3×10^{-4}, and weight decay 10^{-4}. At round n, prior training continued for 200 epochs on $\mathcal{D}_{n-1}$ using AdamW with batch size 128, learning rate 10^{-3}, and weight decay 10^{-4}. The optimizer was reinitialized at each round. The resulting prior p_θ was held fixed for the remainder of the round.

GFP used $K_{\max} = 16$ because its sequence length $L = 238$ made the RTB trajectory computation at $K_{\max} = 32$ exceed the available memory of a single device. All other tasks used $K_{\max} = 32$.

C.3 Variable-Length AMP Implementation

Let $\mathcal{V}_{\mathrm{AA}}$ denote the vocabulary of 20 standard amino acids. AMP sequences were represented by length-49 arrays. For an AMP sequence of length L, positions $1, \ldots, L$ contained amino-acid tokens and the remaining positions contained $[\mathrm{PAD}]$. The tokens $[\mathrm{PAD}]$ and $[\mathrm{MASK}]$ were not included in $\mathcal{V}_{\mathrm{AA}}$.

The length distribution p_{len} defined in Appendix B.1 was estimated from the empirical sequence-length histogram. Before active learning, the initial distribution $p_{\mathrm{len},0}$ was estimated from $\mathcal{D}_0$ using Laplace smoothing with parameter 1. At round n, p_{len} was refit using the sequence lengths in $\mathcal{D}_{n-1}$.

For each particle, L_{target} was sampled from p_{len} and clipped to $[1, 49]$. The seed was right-padded to length 49. If $L_{\mathrm{target}} > L_{\mathrm{seed}}$, positions $L_{\mathrm{seed}} + 1, \ldots, L_{\mathrm{target}}$ were initialized as $[\mathrm{MASK}]$. Positions after L_{target} were fixed to $[\mathrm{PAD}]$. The sampled target length remained fixed throughout MDD reconstruction, RTB sampling, and Particle Guidance. At active positions, $[\mathrm{PAD}]$ was excluded from the clean-token distribution.

C.4 Task-Specific Parameters

Table 2 summarizes the task-dependent parameters. The coefficient κ controls the uncertainty term in equation 7, δ is the masking probability, and k is the seed rank-weighting coefficient in equation 12. Let η denote the coefficient on diversity during query selection. The *Prefilter* column gives the number of candidates retained after ranking by the acquisition score, and $K_{\max}$ denotes the upper bound on the number of reverse diffusion steps.

The coefficient κ followed the task-dependent settings of GFN-AL-δCS (Kim et al., 2025a). For TF-Bind-8, RNA-A/B/C, and AMP, $\delta = 0.5$ followed the masking probability used by GFN-AL-δCS for sequences of length at most 50 (Kim et al., 2025a). For Pab1 and E4B, $\delta = 10/L$ corresponded to the upper end of the corruption range used by ProSpero for these tasks (Kmicikiewicz et al., 2025), giving $10/75$ and $10/102$, respectively. The seed rank coefficient was $k = 0.01$ for all tasks except GFP, which used $k = 0.001$.

C.5 RTB Optimization

For AMP, the expectation in equation 19 was estimated using 512 samples from $p_{\text{len},0}$. Positions fixed to [PAD] were excluded from the token-entropy term.

At each active-learning round, the posterior parameters were initialized from the current prior parameters, $\psi \leftarrow \theta$. RTB optimization ran for 2,000 epochs with batch size 32 and learning rate 3×10^{-4}.

The normalizing factor $Z_\omega(\tilde{x})$ in equation 20 was represented by a two-layer MLP with hidden dimension 64. The normalization model was reinitialized at each round and trained jointly with the posterior using learning rate 10^{-3}. Its initial output was fitted by least squares to β_R times the mean normalized reward in the initialization batch.

Fresh trajectories were sampled with clean-token temperature 2.0 followed by uniform token mixing with probability 0.001. Temperature scaling and uniform mixing were applied only to the clean-token distribution at masked positions and did not alter reveal probabilities. Particle Guidance was not applied during RTB optimization.

The RTB loss was evaluated using unguided trajectory probabilities at temperature 1.0 under the posterior and the fixed prior. Prior log probabilities were cached for replayed trajectories. The replay buffer had capacity 2,048, and replayed trajectories formed 50% of each training batch. The replay buffer was cleared after each active-learning round. Trajectories sampled during RTB optimization were not reused for final candidate generation.

For GFP, likelihood gradients were accumulated over subsets of reverse diffusion steps to reduce peak memory usage while retaining the complete RTB residual.

C.6 Particle-Guided Candidate Generation

Final candidate generation used clean-token temperature 1.0 without uniform token mixing. At each active-learning round, 67,200 raw candidates were generated in 21 batches of 3,200 particles. The particle count defined in Appendix B.3.1 was therefore $N_{\text{PG}} = 3{,}200$. Seed sequences were sampled independently for each particle, and AMP target lengths were sampled independently from p_{len}.

The neighbor temperature in equation 23 was $\tau_s = 0.1$, and the target mean KL divergence in Appendix B.3.2 was $t_{\text{KL}} = 0.03$.

At each reverse diffusion step, one value of β_{PG} was calibrated for the current sampling batch. Let $\beta_{\max}$ denote the largest guidance coefficient allowed during calibration; we set $\beta_{\max} = 65{,}536$. The search started from the interval $[0, 2]$, and the upper endpoint was doubled until the target KL divergence was bracketed or $\beta_{\max}$ was reached. When the target was bracketed, 12 bisection iterations were performed, followed by linear interpolation within the final interval. If the target remained unattainable at $\beta_{\max}$, the guidance coefficient was set to $\beta_{\max}$.

Calibration was skipped when $t_{\text{KL}} = 0$ or when no masked positions remained. For AMP, positions fixed to [PAD] were excluded from both Particle Guidance and KL calibration.

C.7 Candidate Selection

Duplicate candidates and sequences already contained in $\mathcal{D}_{n-1}$ were removed before query selection. The remaining candidates were ranked by the acquisition score $a(x)$, and the task-specific number shown in the *Prefilter* column of Table 2 was retained.

Let $\mathcal{C}$ denote the retained candidate set, and let $\mathcal{S}$ denote the set of candidates already selected for the current query batch. Let $s_{\mathcal{C}}$ denote the standard deviation of the acquisition scores $\{a(x) : x \in \mathcal{C}\}$. For fixed-length tasks, let $d(x, x')$ denote the Hamming distance between two sequences divided by their common sequence length. For AMP, $d(x, x')$ denotes the Hamming distance between the length-49 padded representations divided by 49, so an amino-acid–[PAD] mismatch contributes to the distance.

The first candidate was the sequence in $\mathcal{C}$ with the largest acquisition score. Each subsequent candidate was selected as

$$x^* = \arg\max_{x \in \mathcal{C} \setminus \mathcal{S}} \left[a(x) + \eta s_{\mathcal{C}} \min_{x' \in \mathcal{S}} d(x, x') \right]. \tag{31}$$

Selection continued until the query batch contained B sequences. The value of η was given in Table 2.

D EXPERIMENTAL DETAILS

D.1 TASK DETAILS

Experiments cover 9 task instances across 7 biological sequence design benchmark families. Dataset and oracle configurations are based on GFN-AL-δCS (Kim et al., 2025a) for TF-Bind-8, RNA, AAV, and GFP, GFN-AL (Jain et al., 2022) for AMP, and ProSpero (Kmicikiewicz et al., 2025) for Pab1 and E4B. All tasks maximize their benchmark oracle scores. DNA and RNA tasks use four-nucleotide alphabets, while peptide and protein tasks use the 20 standard amino acids.

For each task, x_{start} denotes the sequence with the highest fitness in the initial measured dataset $\mathcal{D}_0$. We report its fitness $f(x_{\text{start}})$ and the mean distance between x_{start} and the sequences in $\mathcal{D}_0$. The mean distance characterizes the spread of the initial dataset around its highest-fitness sequence. Distances are computed as described in Appendix D.3.

1. **TF-Bind-8.** The task optimizes length-8 DNA sequences for binding to the benchmark target transcription factor. Oracle scores are obtained from the experimentally measured binding landscape (Barrera et al., 2016). The provided low-fitness initial dataset contains $N_0 = 32{,}898$ sequences with $f(\mathbf{x}_{\text{start}}) = 0.4393$ and a mean distance of 5.44 from $\mathbf{x}_{\text{start}}$.
2. **RNA.** RNA-A, RNA-B, and RNA-C optimize length-14 sequences against three distinct target RNA sequences. Each task uses a provided initial dataset of $N_0 = 5{,}000$ sequences (Kim et al., 2023). Fitness is computed from ViennaRNA (Lorenz et al., 2011) duplex binding energies, normalized by a length-scaled complementary-sequence reference. The three tasks have $f(\mathbf{x}_{\text{start}}) = 0.1291$, 0.1189, and 0.1657, with mean distances of 10.15, 9.59, and 9.78 from $\mathbf{x}_{\text{start}}$.
3. **Antimicrobial Peptide (AMP).** The task optimizes predicted antimicrobial activity for peptides of 8–49 residues. The initial dataset contains $N_0 = 3{,}155$ positive peptides from the benchmark's D1 split, retained after discarding peptides longer than 49 residues. Candidates are scored by the benchmark-provided pretrained classifier associated with the D2 split. The initial dataset has $f(\mathbf{x}_{\text{start}}) = 0.9319$ and a mean distance of 19.79 from $\mathbf{x}_{\text{start}}$. Variable-length generation is described in Appendix C.
4. **Adeno-Associated Virus (AAV).** Sequences represent residues 451–540 of AAV2 VP1 (1-based indexing), giving $L = 90$. The oracle is a normalized additive model of liver tropism based on single-substitution measurements (Ogden et al., 2019). The provided initial dataset contains $N_0 = 15{,}307$ variants with $f(\mathbf{x}_{\text{start}}) = 0.5000$ and a mean distance of 5.05 from $\mathbf{x}_{\text{start}}$.
5. **Poly(A)-Binding Protein (Pab1).** The task is based on deep mutational scanning of the RNA-recognition motif of *Saccharomyces cerevisiae* Pab1 (Melamed et al., 2013). Sequences have length $L = 75$, and the fixed initial split contains $N_0 = 10{,}000$ sequences with $f(\mathbf{x}_{\text{start}}) = 0.8429$ and a mean distance of 3.95 from $\mathbf{x}_{\text{start}}$. Initial sequences and generated candidates are scored by the same task-specific TAPE-based oracle provided by ProSpero (Rao et al., 2019; Kmicikiewicz et al., 2025).
6. **Ubiquitination Factor E4B (E4B).** The task optimizes variants of the E4B ubiquitination factor (Starita et al., 2013). Sequences have length $L = 102$, and the fixed initial split contains $N_0 = 10{,}000$ sequences with $f(\mathbf{x}_{\text{start}}) = 7.7428$ and a mean distance of 5.42 from $\mathbf{x}_{\text{start}}$. Initial sequences and generated candidates are scored by the same task-specific TAPE-based oracle provided by ProSpero (Rao et al., 2019; Kmicikiewicz et al., 2025).
7. **Green Fluorescent Protein (GFP).** The task optimizes predicted log-fluorescence using a TAPE-based oracle (Rao et al., 2019) for the GFP landscape (Sarkisyan et al., 2016). Sequences have length $L = 238$. The provided initial dataset contains $N_0 = 10{,}200$ variants with $f(\mathbf{x}_{\text{start}}) = 3.5723$ and a mean distance of 42.86 from $\mathbf{x}_{\text{start}}$.

D.2 BASELINE IMPLEMENTATIONS

GFN-AL, GFN-AL-δCS, and PEX use implementations from the GFN-AL-δCS repository (Kim et al., 2025a). AdaLead, DyNA-PPO, CbAS, DbAS, BO, CMA-ES, and TuRBO use FLEXS implementations (Sinai et al., 2020) with the GFN-AL-δCS CNN-ensemble proxy. ProSpero uses its

released implementation, while MLDE is adapted to the active learning loop. Baselines implemented through FLEXS used the shared δ-CS CNN-ensemble proxy configuration. GFN-AL, GFN-AL + δ-CS, and PEX followed the configurations of their implementations in the δ-CS repository. ProSpero retained its released task-specific configuration, while MLDE used the shared CNN ensemble for candidate ranking in our active learning adapter. Our method uses the proxy specified in Appendix C, which differs in target normalization, regularization, early stopping, and the auxiliary classification objective.

1. **GFN-AL-δCS** (Kim et al., 2025a). A GFlowNet reconstructs partially masked measured sequences using uncertainty-dependent masking. The maximum masking probability is 0.05, the uncertainty scaling coefficient is 0.1, and the rank-based proxy reweighting coefficient is $k = 0.01$.
2. **GFN-AL** (Jain et al., 2022). The GFlowNet baseline uses full masking with the masking probability fixed at 1, disabling conservative seed conditioning.
3. **PEX** (Ren et al., 2022). Proximal exploration uses two random mutations and a frontier neighborhood size of five.
4. **AdaLead** (Sinai et al., 2020). Mutation and recombination use a recombination rate of 0.2, a per-position mutation rate of $1/L$, and a selection threshold of $\tau = 0.05$.
5. **DyNA-PPO** (Angermueller et al., 2020). The policy is trained using proximal policy optimization. Both internal fitness-evaluation stages use the surrogate, while oracle queries are handled by the shared active learning runner. In contrast, the original FLEXS implementation alternates between ground-truth and surrogate fitness evaluations within the internal environment.
6. **CbAS** (Brookes et al., 2019). A variational autoencoder operating on one-hot sequences is retrained using the top 20% of available sequences with importance weighting.
7. **DbAS** (Brookes & Listgarten, 2018). The baseline uses the same generator and selection fraction as CbAS, without importance weighting.
8. **Bayesian Optimization (BO)** (Snoek et al., 2012). We used the FLEXS implementation with UCB acquisition, replacing its default expected-improvement acquisition setting.
9. **CMA-ES** (Hansen & Ostermeier, 2001). Search is performed in a continuous representation, with candidates decoded by taking the token-wise argmax at each position.
10. **TuRBO** (Eriksson et al., 2019). Bayesian optimization is performed within adaptive local trust regions, using the same sequence decoding as CMA-ES.
11. **ProSpero** (Kmicikiewicz et al., 2025). The released implementation and configuration are used for pretrained-model-guided sequence exploration.
12. **MLDE** (Tran & Hy, 2025). The Directed Evolution implementation is adapted using 10 optimization steps, a population size of 128, a beam size of four, and a $0.6 : 0.4$ split between random and importance-based masking. Masked tokens are filled using the 35M-parameter ESM-2 model (Lin et al., 2023). Each optimization step corresponds to one active learning round, with candidates ranked by the CNN ensemble and the resulting population evaluated by the oracle.
13. **MDD baseline.** The MDD baseline retains the round-specific MDD prior, the proxy model, the UCB acquisition and the candidate-pool size used by RD3, and disables reward-tilted fine-tuning, particle guidance, the conservative seed initialization and the diversity term in query selection. Generation therefore starts from a fully masked state, and the query batch is the top of the candidate pool by proxy utility.

ProSpero and MLDE are evaluated on AAV, GFP, Pab1, and E4B. CbAS and DbAS are excluded from AMP because their implementations require fixed-length inputs.

D.3 Evaluation protocol

Let $\mathcal{D}_0$ denote the initial dataset. Let $\mathcal{Q}_r$ denote the batch of B sequences submitted in round r to the oracle f, which returns the fitness of a sequence. After N active learning rounds the initial dataset is

pooled with every queried sequence,

$$\mathcal{P} = \mathcal{D}_0 \cup \bigcup_{r=1}^{N} \mathcal{Q}_r,$$

and the $K = 128$ highest-ranking sequence-fitness pairs are retained as

$$\mathcal{D}_{\text{best}} = \left\{ \left(x^{(i)}, f(x^{(i)})\right) \right\}_{i=1}^{K}, \qquad f\left(x^{(1)}\right) \geq \cdots \geq f\left(x^{(K)}\right).$$

Records are ranked by the oracle fitness, not by a proxy prediction. Ties are resolved by position in the pool, which places initial-dataset records ahead of queried ones. Following Kim et al. (2025a), the retained records are not deduplicated. A sequence evaluated in two rounds therefore occupies two entries of $\mathcal{D}_{\text{best}}$. Every method uses the same pool and the same K, and every reported value is recomputed from sequence-level outputs.

The sequence distance $d(\cdot, \cdot)$ is the unnormalized Levenshtein distance on TF-Bind-8 and AMP, and the unnormalized Hamming distance elsewhere. Each task thus uses the distance its published value was reported with. For AMP, reported diversity and novelty use unnormalized Levenshtein distance between unpadded peptide sequences, whereas candidate selection uses the encoded Hamming distance defined in Appendix C. Each metric is computed separately for every run. Results are summarized as the mean and sample standard deviation over 5 random seeds. Tables report the value after the final round $N = 10$ unless stated otherwise.

Five metrics are reported, three on the fitness axis and two on the distance axis. The three fitness metrics are the maximum, median and mean of the retained set, as in Kim et al. (2025a).

(i) **Maximum fitness.** The highest fitness a method recovers, and the criterion the headline comparison rests on:

$$\text{MaxFitness}(\mathcal{D}_{\text{best}}) = \max_{x \in \mathcal{D}_{\text{best}}} f(x). \tag{32}$$

(ii) **Mean fitness.** The average fitness across all K records, which judges the batch as a whole rather than its best member:

$$\text{MeanFitness}(\mathcal{D}_{\text{best}}) = \frac{1}{|\mathcal{D}_{\text{best}}|} \sum_{x \in \mathcal{D}_{\text{best}}} f(x). \tag{33}$$

(iii) **Median fitness.** The median fitness over the same records, which separates a batch carried by one outlier from a uniformly good one:

$$\text{MedianFitness}(\mathcal{D}_{\text{best}}) = \operatorname*{median}_{x \in \mathcal{D}_{\text{best}}} f(x). \tag{34}$$

(iv) **Diversity.** The average distance between two retained records. It is large when the retained set spreads across the landscape, and small when it collapses onto one neighborhood:

$$\text{Diversity}(\mathcal{D}_{\text{best}}) = \frac{1}{|\mathcal{D}_{\text{best}}|\left(|\mathcal{D}_{\text{best}}| - 1\right)} \sum_{\substack{x, x' \in \mathcal{D}_{\text{best}} \\ x \neq x'}} d(x, x'). \tag{35}$$

(v) **Novelty.** The average distance from a retained record to the closest sequence in the initial dataset. It measures how far a method travels from the data it was given:

$$\text{Novelty}(\mathcal{D}_{\text{best}}, \mathcal{D}_0) = \frac{1}{|\mathcal{D}_{\text{best}}|} \sum_{x \in \mathcal{D}_{\text{best}}} \min_{x' \in \mathcal{D}_0} d(x, x'). \tag{36}$$

D.4 Ablation and Sensitivity Protocol

The settings below are used for the component ablation in Section 5.4 and the sensitivity analysis in Section 5.5.

Shared configuration. Both studies follow the main experimental protocol. Each run performs 10 active-learning rounds with a query budget of 128 sequences per round. Reported metrics are computed on the 128 retained records defined in Appendix D.3. Since AAV, RNA-C, and E4B are fixed-length tasks, diversity and novelty use Hamming distance. The proxy ensemble, acquisition function, candidate-pool size of 67,200 sequences per round, and query-selection procedure follow Appendix C. The component ablation uses 5 seeds per configuration, and the sensitivity analysis uses 3 seeds per setting.

Component ablation. We compare four configurations on AAV, RNA-C, and E4B. The first uses the MDD prior without RTB posterior learning or Particle Guidance. The second applies Particle Guidance to the prior without RTB posterior learning. The third uses the RTB-trained posterior without Particle Guidance. The fourth is RD3 with both components enabled. All other settings are kept identical to the main experiments. Particle Guidance is used only during candidate generation, and the RTB loss is evaluated with the unguided reverse process.

Sensitivity analysis. We vary the target KL divergence t_{KL}, masking probability δ, and reward coefficient β_R on AAV, one parameter at a time while keeping the other two at their main-experiment values. The evaluated settings are listed in Table 3. Setting $t_{\mathrm{KL}} = 0$ disables Particle Guidance. Since each position is masked independently with probability δ, the main setting $\delta = 0.30$ masks 27 of the 90 AAV positions in expectation.

For the reward coefficient, `auto` denotes the rule in equation 18, where β_R is computed once before active learning from the pretrained prior and the initial dataset. On AAV, the rule gives $\beta_R = 8.47 \pm 0.29$ across the three runs.

Table 3: Sensitivity settings on AAV. Main-experiment values are shown in bold.

Parameter	Symbol	Values
Target KL divergence	t_{KL}	0, **0.03**, 0.1, 0.3, 1.0, 4.0
Masking probability	δ	0.05, 0.15, 0.22, **0.30**, 0.40
Reward coefficient	β_R	2, 4, 8, **`auto`**, 16, 32

D.5 Compute Resources

All RD3 experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB of memory. We used Python 3.7.12, PyTorch 1.12.0, CUDA 11.6, and NumPy 1.21.6.

Representative end-to-end runtimes for 10 active-learning rounds were approximately 2.8 hours on RNA-C and 6.4–7.9 hours on Pab1, AAV, and E4B.

E FULL RESULT

E.1 FULL RESULTS OF BIOLOGICAL SEQUENCE DESIGN

Table 4: Mean fitness of top 128 sequences obtained by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AAV	AMP	E4B	GFP	PAB1	RNA-A	RNA-B	RNA-C	TFBIND8
CMA-ES	0.482 ± 0.000	0.904 ± 0.000	4.607 ± 0.000	3.384 ± 0.000	0.354 ± 0.012	0.672 ± 0.056	0.593 ± 0.129	0.663 ± 0.062	0.906 ± 0.005
BO	0.482 ± 0.003	0.901 ± 0.001	4.269 ± 0.000	3.420 ± 0.070	0.299 ± 0.038	0.936 ± 0.028	0.953 ± 0.017	0.971 ± 0.080	0.971 ± 0.006
TuRBO	0.480 ± 0.000	0.915 ± 0.007	4.269 ± 0.000	3.331 ± 0.000	0.288 ± 0.033	0.947 ± 0.022	0.934 ± 0.013	0.973 ± 0.040	0.971 ± 0.006
DynaPPO	0.480 ± 0.000	0.899 ± 0.000	4.269 ± 0.000	3.331 ± 0.000	0.293 ± 0.017	0.610 ± 0.017	0.613 ± 0.020	0.601 ± 0.020	0.916 ± 0.006
PEX	0.609 ± 0.014	0.958 ± 0.001	7.523 ± 0.114	3.596 ± 0.002	1.259 ± 0.150	0.762 ± 0.024	0.641 ± 0.005	0.763 ± 0.010	0.957 ± 0.002
AdaLead	0.669 ± 0.013	0.936 ± 0.001	7.883 ± 0.045	3.573 ± 0.004	1.558 ± 0.280	0.860 ± 0.016	0.902 ± 0.029	0.883 ± 0.069	0.965 ± 0.004
MLDE	0.518 ± 0.002	–	7.805 ± 0.049	3.583 ± 0.001	0.898 ± 0.004	–	–	–	–
CbAS	0.482 ± 0.001	–	4.269 ± 0.000	3.560 ± 0.001	0.263 ± 0.003	0.596 ± 0.004	0.596 ± 0.013	0.622 ± 0.008	0.927 ± 0.007
DbAS	0.482 ± 0.001	–	4.269 ± 0.000	3.560 ± 0.001	0.288 ± 0.014	0.596 ± 0.017	0.579 ± 0.021	0.626 ± 0.024	0.935 ± 0.009
GFN-AL	0.480 ± 0.000	0.901 ± 0.002	4.269 ± 0.000	3.331 ± 0.000	0.277 ± 0.003	0.513 ± 0.003	0.496 ± 0.011	0.517 ± 0.008	0.787 ± 0.015
GFN-AL-δCS	0.665 ± 0.007	0.937 ± 0.014	7.909 ± 0.080	3.566 ± 0.001	1.467 ± 0.242	0.947 ± 0.010	0.934 ± 0.004	0.983 ± 0.048	0.974 ± 0.003
ProSpero	0.663 ± 0.023	–	7.982 ± 0.021	3.611 ± 0.002	1.348 ± 0.335	–	–	–	–
RD3	0.827 ± 0.029	0.964 ± 0.001	8.039 ± 0.036	3.573 ± 0.005	1.530 ± 0.368	0.973 ± 0.003	0.953 ± 0.009	1.001 ± 0.006	0.962 ± 0.003

Table 5: Median fitness of top 128 sequences obtained by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AAV	AMP	E4B	GFP	PAB1	RNA-A	RNA-B	RNA-C	TFBIND8
CMA-ES	0.481 ± 0.000	0.901 ± 0.000	4.269 ± 0.000	3.410 ± 0.000	0.272 ± 0.022	0.659 ± 0.059	0.576 ± 0.123	0.649 ± 0.061	0.911 ± 0.004
BO	0.480 ± 0.003	0.899 ± 0.001	4.085 ± 0.000	3.439 ± 0.065	0.242 ± 0.052	0.927 ± 0.030	0.950 ± 0.022	0.962 ± 0.078	0.972 ± 0.004
TuRBO	0.478 ± 0.000	0.914 ± 0.007	4.085 ± 0.000	3.378 ± 0.000	0.227 ± 0.043	0.936 ± 0.025	0.929 ± 0.019	0.963 ± 0.037	0.970 ± 0.005
DynaPPO	0.478 ± 0.000	0.897 ± 0.000	4.085 ± 0.000	3.378 ± 0.000	0.235 ± 0.027	0.598 ± 0.018	0.600 ± 0.019	0.589 ± 0.019	0.912 ± 0.008
PEX	0.605 ± 0.015	0.958 ± 0.001	7.511 ± 0.126	3.596 ± 0.002	1.227 ± 0.142	0.755 ± 0.021	0.633 ± 0.007	0.753 ± 0.010	0.957 ± 0.003
AdaLead	0.667 ± 0.013	0.935 ± 0.001	7.870 ± 0.051	3.573 ± 0.004	1.554 ± 0.282	0.852 ± 0.020	0.894 ± 0.031	0.874 ± 0.071	0.965 ± 0.004
MLDE	0.516 ± 0.002	–	7.791 ± 0.051	3.583 ± 0.001	0.900 ± 0.006	–	–	–	–
CbAS	0.480 ± 0.001	–	4.085 ± 0.000	3.567 ± 0.001	0.193 ± 0.005	0.584 ± 0.004	0.586 ± 0.015	0.610 ± 0.010	0.925 ± 0.005
DbAS	0.480 ± 0.001	–	4.085 ± 0.000	3.567 ± 0.001	0.229 ± 0.023	0.587 ± 0.016	0.569 ± 0.023	0.614 ± 0.024	0.931 ± 0.011
GFN-AL	0.478 ± 0.000	0.900 ± 0.003	4.085 ± 0.000	3.378 ± 0.000	0.212 ± 0.005	0.498 ± 0.003	0.481 ± 0.007	0.501 ± 0.008	0.778 ± 0.016
GFN-AL-δCS	0.663 ± 0.007	0.936 ± 0.015	7.902 ± 0.081	3.565 ± 0.002	1.455 ± 0.238	0.939 ± 0.009	0.929 ± 0.004	0.972 ± 0.048	0.973 ± 0.003
ProSpero	0.660 ± 0.023	–	7.977 ± 0.024	3.611 ± 0.002	1.344 ± 0.333	–	–	–	–
RD3	0.824 ± 0.030	0.964 ± 0.001	8.033 ± 0.036	3.572 ± 0.005	1.518 ± 0.369	0.966 ± 0.005	0.947 ± 0.008	0.990 ± 0.004	0.962 ± 0.004

Table 6: Diversity of top 128 sequences obtained by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AAV	AMP	E4B	GFP	PAB1	RNA-A	RNA-B	RNA-C	TFBIND8
CMA-ES	4.148 ± 0.000	24.508 ± 1.157	4.850 ± 0.000	58.299 ± 0.000	29.573 ± 3.742	4.342 ± 0.297	4.888 ± 0.505	4.276 ± 0.326	3.031 ± 0.151
BO	5.429 ± 1.433	17.922 ± 1.032	4.896 ± 0.000	58.057 ± 6.124	35.260 ± 12.946	2.894 ± 0.264	3.309 ± 0.575	2.661 ± 1.442	2.546 ± 0.352
TuRBO	4.536 ± 0.000	20.396 ± 10.390	4.896 ± 0.000	62.955 ± 0.000	24.991 ± 14.178	6.225 ± 0.649	5.480 ± 1.232	6.706 ± 0.841	2.934 ± 0.242
DynaPPO	4.536 ± 0.000	14.125 ± 0.000	4.896 ± 0.000	62.955 ± 0.000	41.292 ± 9.910	8.634 ± 0.112	8.440 ± 0.210	8.826 ± 0.192	3.803 ± 0.067
PEX	8.526 ± 1.701	6.806 ± 0.590	5.213 ± 0.721	10.111 ± 0.812	6.226 ± 2.122	4.069 ± 0.823	3.292 ± 0.173	3.799 ± 0.549	3.791 ± 0.048
AdaLead	8.011 ± 2.443	14.546 ± 0.988	6.044 ± 0.934	24.998 ± 11.744	4.515 ± 0.944	4.503 ± 1.169	3.908 ± 0.865	3.167 ± 0.120	3.710 ± 0.073
MLDE	4.031 ± 0.208	–	4.955 ± 0.256	6.460 ± 0.683	4.608 ± 0.325	–	–	–	–
CbAS	4.232 ± 0.095	–	4.896 ± 0.000	14.178 ± 0.493	14.901 ± 6.228	9.083 ± 0.161	8.686 ± 0.265	8.590 ± 0.942	4.054 ± 0.105
DbAS	4.232 ± 0.095	–	4.896 ± 0.000	14.178 ± 0.493	37.964 ± 4.336	8.811 ± 0.661	8.695 ± 1.047	8.551 ± 0.997	3.910 ± 0.135
GFN-AL	4.536 ± 0.000	23.354 ± 5.338	4.896 ± 0.000	62.955 ± 0.000	30.642 ± 1.690	9.908 ± 0.055	9.936 ± 0.095	10.106 ± 0.021	4.818 ± 0.047
GFN-AL-δCS	11.296 ± 0.968	21.652 ± 10.714	10.104 ± 2.105	59.178 ± 6.064	6.181 ± 1.050	6.442 ± 0.587	5.644 ± 0.343	6.493 ± 1.958	2.183 ± 0.230
ProSpero	6.019 ± 0.838	–	4.046 ± 0.652	10.395 ± 1.573	3.885 ± 0.250	–	–	–	–
RD3	14.546 ± 2.242	20.661 ± 1.217	5.902 ± 0.890	75.359 ± 6.277	4.287 ± 0.759	6.739 ± 0.136	6.645 ± 0.375	7.678 ± 0.390	3.637 ± 0.028

Table 7: Novelty of top 128 sequences obtained by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AAV	AMP	E4B	GFP	PAB1	RNA-A	RNA-B	RNA-C	TFBIND8
CMA-ES	0.000 ± 0.000	7.970 ± 1.218	0.000 ± 0.000	0.000 ± 0.000	15.398 ± 2.582	6.384 ± 0.290	6.008 ± 1.058	5.827 ± 0.293	0.705 ± 0.086
BO	0.706 ± 1.073	2.286 ± 0.700	0.000 ± 0.000	8.467 ± 11.844	25.698 ± 16.156	7.303 ± 0.303	7.766 ± 0.143	6.286 ± 0.179	1.398 ± 0.211
TuRBO	0.000 ± 0.000	27.445 ± 4.746	0.000 ± 0.000	0.000 ± 0.000	21.403 ± 18.552	7.513 ± 0.067	7.828 ± 0.046	6.602 ± 0.073	1.466 ± 0.161
DynaPPO	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	23.834 ± 7.205	6.167 ± 0.128	6.412 ± 0.214	5.739 ± 0.174	1.298 ± 0.026
PEX	6.653 ± 1.089	34.386 ± 0.379	2.703 ± 0.285	8.127 ± 1.692	6.227 ± 1.925	6.770 ± 0.167	5.539 ± 0.079	5.589 ± 0.373	1.131 ± 0.017
AdaLead	8.920 ± 0.431	32.147 ± 0.390	4.842 ± 1.239	3.864 ± 0.600	8.844 ± 1.232	7.050 ± 0.098	7.558 ± 0.172	5.987 ± 0.293	1.253 ± 0.030
MLDE	3.237 ± 0.113	–	3.730 ± 0.338	4.838 ± 0.322	3.534 ± 0.138	–	–	–	–
CbAS	0.095 ± 0.028	–	0.000 ± 0.000	1.698 ± 0.076	6.398 ± 3.587	6.119 ± 0.097	6.231 ± 0.031	5.636 ± 0.162	1.198 ± 0.017
DbAS	0.095 ± 0.028	–	0.000 ± 0.000	1.698 ± 0.076	25.752 ± 7.979	6.083 ± 0.121	6.141 ± 0.139	5.664 ± 0.171	1.275 ± 0.031
GFN-AL	0.000 ± 0.000	6.252 ± 4.526	0.000 ± 0.000	0.000 ± 0.000	15.730 ± 1.103	5.714 ± 0.083	5.778 ± 0.122	5.303 ± 0.081	1.114 ± 0.025
GFN-AL-δCS	10.233 ± 0.918	26.720 ± 3.021	8.578 ± 2.029	20.036 ± 2.540	10.989 ± 1.360	7.406 ± 0.073	7.661 ± 0.072	6.494 ± 0.094	1.853 ± 0.237
ProSpero	14.722 ± 1.615	–	7.787 ± 1.163	39.191 ± 4.393	11.159 ± 1.752	–	–	–	–
RD3	20.075 ± 1.848	24.709 ± 0.738	6.809 ± 0.830	42.703 ± 3.328	6.987 ± 0.159	7.573 ± 0.038	7.814 ± 0.034	6.566 ± 0.043	1.381 ± 0.033

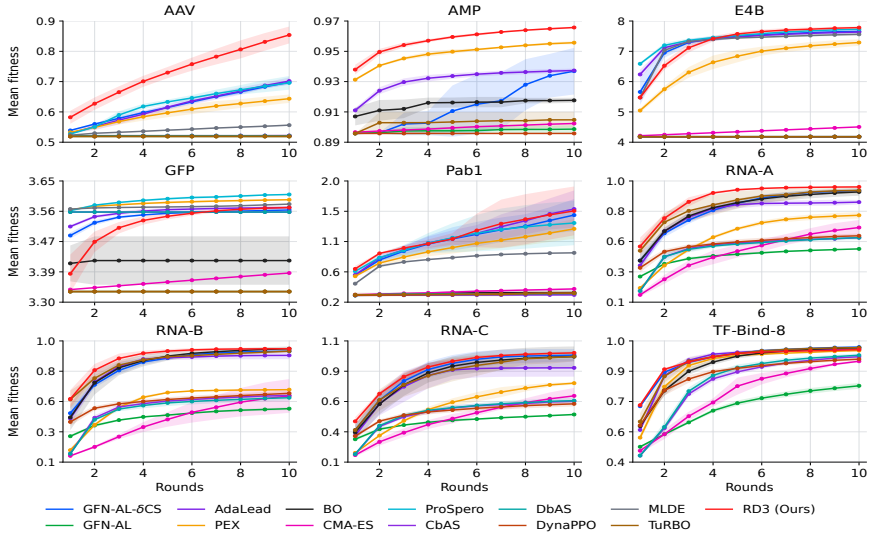


Figure 6: Mean fitness achieved over 10 active learning rounds across 9 biological sequence design tasks. Solid lines represent the mean across 5 runs, and shaded regions indicate the standard deviation. RD3 (Ours) is shown in red.

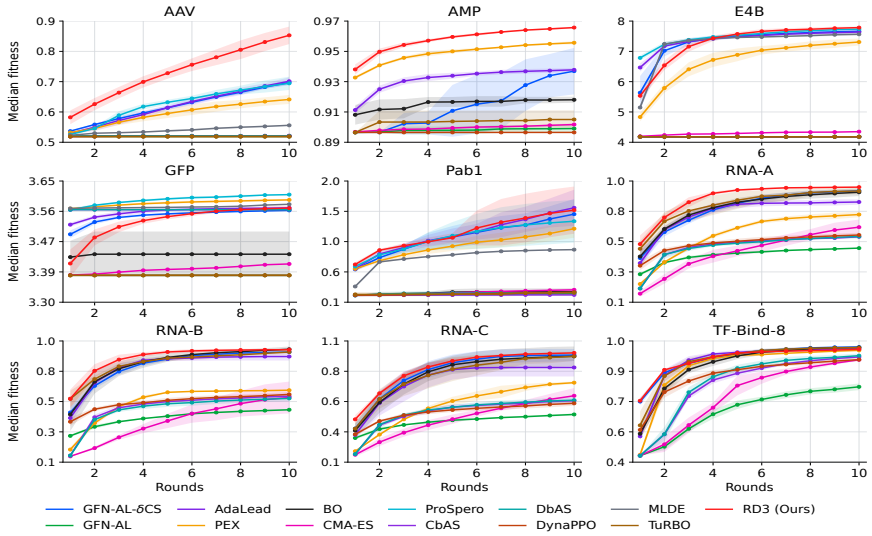


Figure 7: Median fitness achieved over 10 active learning rounds across 9 biological sequence design tasks. Solid lines represent the mean across 5 runs, and shaded regions indicate the standard deviation. RD3 (Ours) is shown in red.

E.2 Result of Fitness–Novelty Trade-off

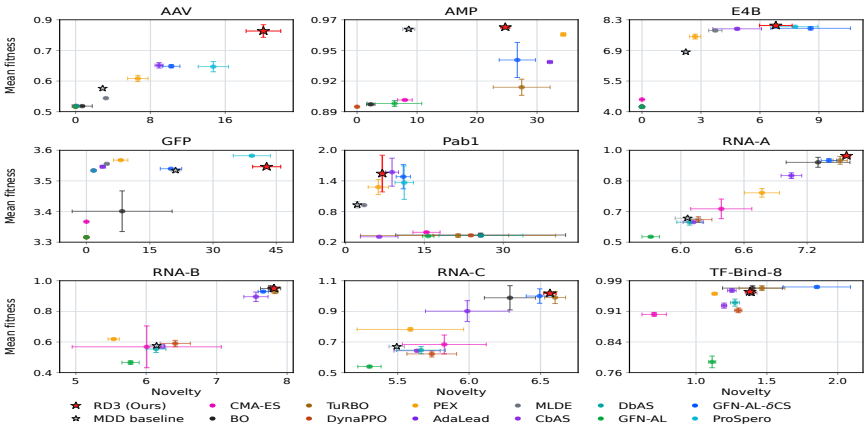


Figure 8: Mean fitness of the retained set against novelty after 10 active learning rounds. Each point is the mean over 5 runs, and the horizontal and vertical bars give the standard deviation in novelty and in mean fitness.

E.3 Full results of Ablation Study

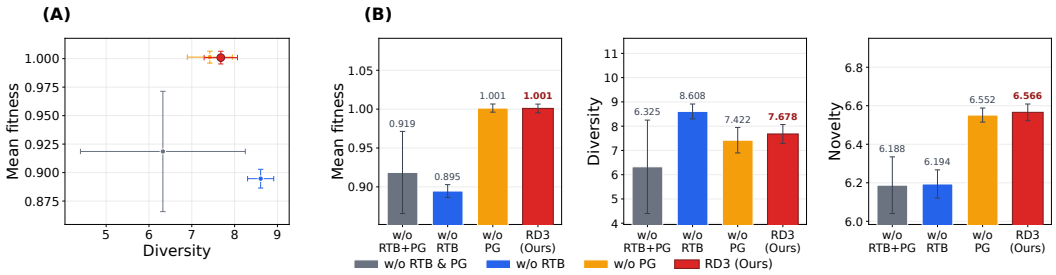


Figure 9: Component ablation on RNA-C. (A) Mean fitness of the retained set against diversity for the four configurations. (B) The same four configurations reported one metric at a time, for mean fitness, diversity and novelty. Each point and each bar is the mean over 5 runs, and the error bars give the standard deviation.

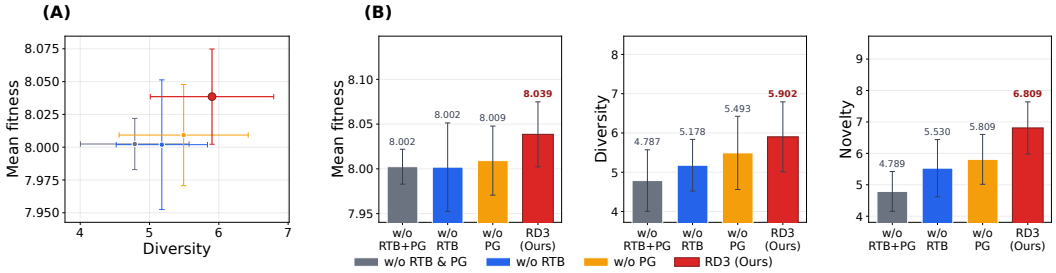


Figure 10: Component ablation on E4B. (A) Mean fitness of the retained set against diversity for the four configurations. (B) The same four configurations reported one metric at a time, for mean fitness, diversity and novelty. Each point and each bar is the mean over 5 runs, and the error bars give the standard deviation.

Table 8: Maximum fitness of the top 128 sequences under each ablation variant. Reported values are the mean and standard deviation over 5 runs.

Ablation variant	AAV	RNA-C	E4B
w/o RTB & PG	0.733 ± 0.032	1.064 ± 0.077	8.128 ± 0.033
w/o RTB	0.715 ± 0.010	1.095 ± 0.044	8.111 ± 0.048
w/o PG	0.864 ± 0.022	1.116 ± 0.000	8.115 ± 0.021
RD3 (Ours)	$\mathbf{0.879 \pm 0.023}$	$\mathbf{1.115 \pm 0.002}$	$\mathbf{8.136 \pm 0.023}$

Table 9: Mean fitness of the top 128 sequences under each ablation variant. Reported values are the mean and standard deviation over 5 runs.

Ablation variant	AAV	RNA-C	E4B
w/o RTB & PG	0.699 ± 0.028	0.919 ± 0.053	8.002 ± 0.020
w/o RTB	0.675 ± 0.006	0.895 ± 0.008	8.002 ± 0.049
w/o PG	0.824 ± 0.020	1.001 ± 0.005	8.009 ± 0.039
RD3 (Ours)	$\mathbf{0.827 \pm 0.029}$	$\mathbf{1.001 \pm 0.006}$	$\mathbf{8.039 \pm 0.036}$

Table 10: Median fitness of the top 128 sequences under each ablation variant. Reported values are the mean and standard deviation over 5 runs.

Ablation variant	AAV	RNA-C	E4B
w/o RTB & PG	0.696 ± 0.028	0.912 ± 0.055	7.997 ± 0.022
w/o RTB	0.672 ± 0.006	0.881 ± 0.009	7.997 ± 0.047
w/o PG	0.822 ± 0.020	0.991 ± 0.004	8.003 ± 0.040
RD3 (Ours)	$\mathbf{0.824 \pm 0.030}$	$\mathbf{0.990 \pm 0.004}$	$\mathbf{8.033 \pm 0.036}$

Table 11: Diversity of the top 128 sequences under each ablation variant. Reported values are the mean and standard deviation over 5 runs.

Ablation variant	AAV	RNA-C	E4B
w/o RTB & PG	5.883 ± 0.460	6.325 ± 1.925	4.787 ± 0.784
w/o RTB	7.303 ± 0.599	8.608 ± 0.305	5.178 ± 0.659
w/o PG	9.717 ± 1.075	7.422 ± 0.525	5.493 ± 0.932
RD3 (Ours)	$\mathbf{14.546 \pm 2.242}$	$\mathbf{7.678 \pm 0.390}$	$\mathbf{5.902 \pm 0.890}$

Table 12: Novelty of the top 128 sequences under each ablation variant. Reported values are the mean and standard deviation over 5 runs.

Ablation variant	AAV	RNA-C	E4B
w/o RTB & PG	7.355 ± 1.240	6.188 ± 0.147	4.789 ± 0.634
w/o RTB	6.947 ± 0.185	6.194 ± 0.073	5.530 ± 0.909
w/o PG	15.792 ± 1.427	6.552 ± 0.036	5.809 ± 0.794
RD3 (Ours)	$\mathbf{20.075 \pm 1.848}$	$\mathbf{6.566 \pm 0.043}$	$\mathbf{6.809 \pm 0.830}$

E.4 Full results of Sensitivity Analysis

Table 13: Per-grid-point sensitivity results on AAV at the final active learning round. Values are mean ± standard deviation over the available seeds. RD3 headline settings are bold.

Parameter	Setting	Mean fitness	Median fitness	Maximum fitness	Diversity	Novelty
t_{KL}	0	0.824 ± 0.020	0.822 ± 0.020	0.864 ± 0.022	9.717 ± 1.075	15.792 ± 1.427
t_{KL}	**0.03**	**0.827 ± 0.029**	**0.824 ± 0.030**	**0.879 ± 0.023**	**14.546 ± 2.242**	**20.075 ± 1.848**
t_{KL}	0.1	0.780 ± 0.011	0.776 ± 0.009	0.834 ± 0.020	20.792 ± 1.487	22.094 ± 1.137
t_{KL}	0.3	0.733 ± 0.018	0.727 ± 0.017	0.798 ± 0.020	26.306 ± 0.948	24.573 ± 1.273
t_{KL}	1.0	0.631 ± 0.003	0.626 ± 0.004	0.728 ± 0.022	30.148 ± 0.505	22.586 ± 0.843
t_{KL}	4.0	0.493 ± 0.001	0.489 ± 0.002	0.600 ± 0.026	15.866 ± 1.121	6.802 ± 0.716
δ	0.05	0.787 ± 0.007	0.785 ± 0.007	0.822 ± 0.006	5.122 ± 0.122	13.300 ± 1.690
δ	0.15	0.824 ± 0.022	0.822 ± 0.022	0.862 ± 0.029	9.730 ± 1.736	17.714 ± 1.741
δ	0.22	0.800 ± 0.033	0.797 ± 0.034	0.849 ± 0.023	13.702 ± 1.294	18.776 ± 1.281
δ	**0.30**	**0.827 ± 0.029**	**0.824 ± 0.030**	**0.879 ± 0.023**	**14.546 ± 2.242**	**20.075 ± 1.848**
δ	0.40	0.759 ± 0.065	0.757 ± 0.066	0.814 ± 0.071	13.053 ± 4.847	14.763 ± 6.440
β_R	2	0.689 ± 0.019	0.687 ± 0.019	0.731 ± 0.032	7.928 ± 0.631	7.812 ± 0.706
β_R	4	0.725 ± 0.020	0.722 ± 0.020	0.767 ± 0.022	8.281 ± 1.272	9.495 ± 1.108
β_R	8	0.800 ± 0.011	0.796 ± 0.011	0.863 ± 0.022	14.421 ± 1.422	17.799 ± 1.346
β_R	**Auto**	**0.827 ± 0.029**	**0.824 ± 0.030**	**0.879 ± 0.023**	**14.546 ± 2.242**	**20.075 ± 1.848**
β_R	16	0.774 ± 0.014	0.769 ± 0.014	0.841 ± 0.004	20.921 ± 1.552	23.224 ± 2.242
β_R	32	0.728 ± 0.006	0.724 ± 0.006	0.790 ± 0.024	22.047 ± 0.380	21.440 ± 0.603

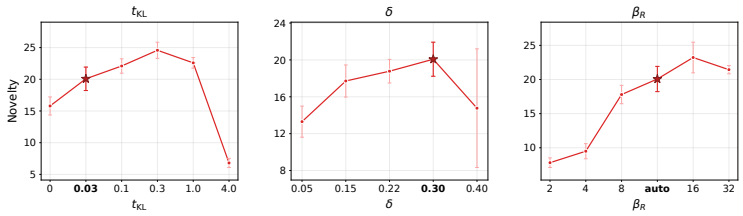


Figure 11: Novelty along the three sensitivity grids on AAV. Each column varies one hyperparameter while keeping the others fixed. Stars indicate the RD3 configuration. The automatically selected reward strength (β_R) is 8.47 ± 0.29 across runs. Results are reported as mean ± standard deviation over three runs.