DiMOS: Doob-Guided Inference-Time Multi-Objective Search for Scientific Design

Anonymous authors
Paper under double-blind review

Abstract

Scientific design often requires jointly satisfying multiple objectives and constraints. Pretrained masked diffusion models provide a generative foundation for this task, but fine-tuning them to meet these objectives and constraints incurs additional training costs, motivating inference-time guidance with frozen models. However, such guidance faces two challenges: pass-or-fail constraints and black-box reward models may provide no useful gradients, while jointly satisfying multiple requirements can leave a small feasible region, making feasible designs difficult to find within a limited inference budget. To address these challenges, we introduce DiMOS, a training-free framework for multi-objective scientific design. Using joint rewards from candidate completions, DiMOS performs approximate Doob-guided local resampling without requiring reward gradients. To allocate computation efficiently, it uses budget-efficient trajectory search to focus computation on promising continuations. Across six DNA, protein, and RNA tasks, DiMOS attains the highest joint success rate at comparable generation times, up to $1.98\times$ the strongest baseline on DNA and protein, while maintaining high sequence uniqueness and naturalness. Our code is available at `https://anonymous.4open.science/r/DiMOS/`.

1 Introduction

Scientific sequence design is a central task in synthetic biology and therapeutic development (Zrimec et al., 2022; Cao et al., 2022). In practice, however, scientists often need to satisfy several requirements simultaneously rather than improve a single property in isolation (Stanton et al., 2022; Gruver et al., 2023). For example, regulatory DNA should combine target-cell activity with limited off-target activity (Gosai et al., 2024), while therapeutic mRNA requires both efficient translation and sufficient stability (Leppek et al., 2022). We therefore focus on *multi-objective scientific sequence design*, seeking candidate sequences that jointly meet multiple objectives and constraints.

Masked diffusion models provide a generative foundation for this task by learning sequence distributions through the reconstruction of masked tokens (Sahoo et al., 2024; Wang et al., 2025). However, sampling from a learned distribution does not ensure that designs meet the specified objectives (Li et al., 2025). Fine-tuning-based methods address this gap by updating pretrained models toward task-specific rewards (Black et al., 2024; Clark et al., 2024; Wang et al., 2025; Han et al., 2026; Domingo-Enrich et al., 2025), but the added training becomes a burden as objectives or thresholds change (Li et al., 2025). This motivates inference-time guidance that reuses a frozen generative prior while steering generation toward different combinations of objectives and constraints. Such guidance needs one signal to follow, so we assign each candidate a joint reward $R(x)$ that is high only when every requirement is met.

Yet guiding a frozen model with such a reward presents two challenges:

- ❶ ***How can we guide generation when R is non-differentiable?*** Scientific design often requires improving one property while enforcing hard constraints on others. Even when reward models are differentiable, pass-or-fail constraints provide no useful gradient signal for correcting constraint violations. The reward model itself can also be non-differentiable: black-box tools such as ViennaRNA assess RNA folding without directly providing sequence gradients (Lorenz et al.,

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.

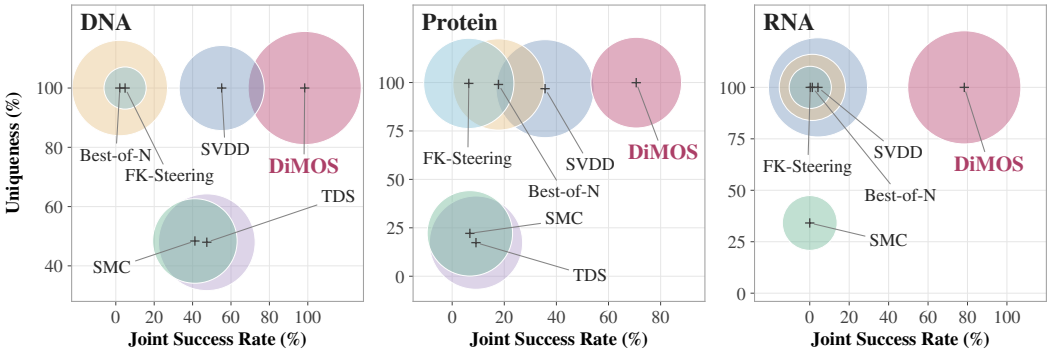


Figure 1: **Joint success and uniqueness.** Bubble area encodes within-task normalized 3-mer correlation (DNA), scRMSD < 2 Å rate (protein), and Optimus-MRL (RNA). Three-seed means.

2011). Gradient-based inference-time methods therefore struggle to follow such a reward (Wu et al., 2023; Nisonoff et al., 2025).

- ❷ ***How can we effectively search for jointly feasible designs within a given inference budget?*** Gradient-free methods handle non-differentiable feedback but often require extensive sampling and repeated reward evaluations (Li et al., 2025; Singhal et al., 2025). Meanwhile, simultaneously satisfying multiple objectives and constraints can leave only a small feasible region, particularly when objectives conflict, further increasing the search effort required (Stanton et al., 2022). Efficiently allocating computation to discover jointly feasible candidates within a limited inference budget is therefore another key challenge.

To address these challenges, we propose DIMOS, a training-free framework for multi-objective scientific design with frozen masked diffusion models (Fig. 2). Its approximate Doob-guided local resampling steers generation using reward evaluations alone, accommodating non-differentiable feedback. Building on this local guidance, budget-efficient trajectory search directs computation toward promising continuations to find jointly feasible designs. Our three contributions are:

- **Framework.** We propose DIMOS, a Doob-guided framework for inference-time multi-objective design with frozen masked diffusion models. It supports black-box objectives and hard constraints through evaluations of R, keeping the generative model frozen and requiring reward values alone.
- **Algorithm.** We develop a budget-efficient trajectory search algorithm that combines stage-wise branching with reward-guided refinement. It allocates computation across local candidate comparisons and promising continuations to improve joint success within a given inference budget.
- **Effectiveness.** Across six multi-objective tasks spanning DNA, protein, and RNA design (Fig. 1), DIMOS attains the highest joint success rate under matched wall-clock budgets, up to $1.98\times$ the strongest baseline on DNA and protein, while maintaining sequence naturalness.

2 Related Work

Multi-Objective Scientific Design. Multi-objective scientific design seeks sequences with several desired properties. Discrete diffusion supports regulatory DNA design (Sarkar et al., 2024). Constrained reinforcement learning adapts generative models to optimize target-cell activity while controlling off-target effects (Chen et al., 2025b). Other approaches explore Pareto trade-offs through multi-objective Bayesian optimization (Stanton et al., 2022), peptide tree search (Tang et al., 2025), or rank-directional flow guidance (Chen et al., 2025a). Pareto optimality alone, however, does not ensure simultaneous satisfaction of application thresholds. Our focus is finding designs that jointly meet specified property thresholds through inference-time search.

Training-Free Adaptation. Training-free methods steer a frozen generative prior at sampling time using reward feedback. (1) *Gradient guidance* modifies reverse transitions using reward gradients or gradient-based approximations (Nisonoff et al., 2025; Schiff et al., 2025). (2) *Particle methods* reweight and resample trajectories, sometimes using gradient-guided proposals (Wu et al., 2023; Singhal et al., 2025). (3) *Evaluation-based guidance* ranks or resamples candidate continuations

using reward values or estimated future rewards (Guo et al., 2024; Li et al., 2025), with iterative refinement wrapping single-shot guidance in an outer noising-and-denoising loop (Uehara et al., 2025a). That outer loop is orthogonal to the per-step mechanisms studied here and could wrap any of them. Related probabilistic formulations connect adaptation to Doob's h-transform, soft value functions, and posterior sampling (Uehara et al., 2025b). These include learned corrections or samplers (Denker et al., 2024; Rector-Brooks et al., 2025), as well as training-free inference for image editing (Nguyen et al., 2025) and adaptation with generic rewards (Zhu et al., 2026). DIMOS uses reward evaluations alone for approximate Doob-guided local resampling, then concentrates computation on promising continuations via shared prefixes and branch selection.

3 PRELIMINARIES

We introduce discrete diffusion (Austin et al., 2021) in Sec. 3.1 and Doob's h-transform in Sec. 3.2.

3.1 DISCRETE DIFFUSION

Let $\mathcal{X} = \mathcal{V}^N$ be the space of length-N sequences over a finite alphabet $\mathcal{V}$. A forward continuous-time Markov chain $(Y_t)_{t\in[0,T]}$ corrupts $Y_0 \sim p_{\mathrm{data}}$ on a finite state space $\mathcal{S} \supseteq \mathcal{X}$. We use the column convention: for $y \neq x$, $Q_t(y, x) \geq 0$ is the jump rate from x to y. The marginal probability vector p_t evolves as

$$\frac{dp_t}{dt} = Q_t p_t, \qquad Q_t(x, x) = -\sum_{y\neq x} Q_t(y, x), \qquad p_0 = p_{\mathrm{data}}. \tag{1}$$

The corruption process is chosen so that p_T approaches a tractable reference distribution.

Writing denoising time as $u = T - t$, define the reverse-time process $\widetilde{Y}_u := Y_{T-u}$, so that denoising runs from $u = 0$ to $u = T$, corresponding to the forward-time direction $t = T$ to $t = 0$. Its exact off-diagonal generator is

$$\widetilde{Q}_u(y, x) = Q_{T-u}(x, y)\frac{p_{T-u}(y)}{p_{T-u}(x)}, \qquad y \neq x, \tag{2}$$

for states with $p_{T-u}(x) > 0$. Thus, starting from $\widetilde{Y}_0 \sim p_T$, the reverse process progressively removes the corruption introduced by the forward process and recovers $\widetilde{Y}_T \sim p_0 = p_{\mathrm{data}}$.

The reverse dynamics in Eq. (2) depend on the density ratios $p_t(y)/p_t(x)$, which characterize how the reverse process should move between discrete states (Lou et al., 2024).

Masked Discrete Diffusion. A commonly used specialization of discrete diffusion for sequence generation is a masked discrete diffusion model (Sahoo et al., 2024; Shi et al., 2024). For sequence generation, we augment the token alphabet $\mathcal{V}$ with a mask symbol $\mathtt{m} \notin \mathcal{V}$ and let $\widetilde{\mathcal{V}} = \mathcal{V} \cup \{\mathtt{m}\}$ and $\mathcal{S} = \widetilde{\mathcal{V}}^N$. Masked discrete diffusion independently replaces each clean token by $\mathtt{m}$ at a time-dependent rate λ_t (Sahoo et al., 2024; Shi et al., 2024). Writing $\alpha_t = \exp(-\int_0^t \lambda_r\, dr)$ for the probability that a coordinate remains unmasked by time t, the forward transition from $x_0 \in \mathcal{X}$ is

$$q_{0,t}(x \mid x_0) = \prod_{j=1}^{N} \left[\alpha_t \mathbf{1}\{x^j = x_0^j\} + (1 - \alpha_t)\mathbf{1}\{x^j = \mathtt{m}\}\right]. \tag{3}$$

Thus, the forward process progressively replaces clean tokens by masks and never changes one clean token directly into another. As $\alpha_T \to 0$, p_T approaches this fully masked reference distribution $\delta_{\mathtt{m}^N}$.

For a masked coordinate $x^j = \mathtt{m}$, let $x^{j\leftarrow a}$ denote the state obtained by replacing that coordinate with $a \in \mathcal{V}$. Under the masking kernel, $\frac{p_t(x^{j\leftarrow a})}{p_t(x)} = \frac{\alpha_t}{1-\alpha_t}\mathbb{P}(Y_0^j = a \mid Y_t = x)$. Thus the reverse dynamics are parameterized by the clean-token posterior. We approximate this posterior with $\mu^j_{\theta,t}(\cdot \mid x) \in \Delta(\mathcal{V})$, where $\Delta(\mathcal{V})$ is the simplex over $\mathcal{V}$, yielding $\widehat{Q}_{\theta,T-t}\big(x^{j\leftarrow a}, x\big) = \lambda_t \frac{\alpha_t}{1-\alpha_t}\mu^j_{\theta,t}(a \mid x)$.

At inference time, we apply a token temperature $\eta > 0$ to the denoiser distribution and define

$$\mu^{j,\eta}_{\theta,t}(a \mid x) = \frac{\mu^j_{\theta,t}(a \mid x)^{1/\eta}}{\sum_{a'\in\mathcal{V}} \mu^j_{\theta,t}(a' \mid x)^{1/\eta}}. \tag{4}$$

Thus, $\eta = 1$ recovers the denoiser distribution, while $\eta < 1$ sharpens it and $\eta > 1$ flattens it.

The reverse process is simulated on a finite denoising grid $0 = t_0 < t_1 < \cdots < t_L = T$. Discretizing the reverse dynamics over each interval $[t_{\ell-1}, t_\ell]$ gives a transition kernel $P_{\theta,t_\ell}(y \mid x) := \mathbb{P}_\theta\big(X_{t_{\ell-1}} = y \mid X_{t_\ell} = x\big)$. For masked diffusion, this transition progressively reveals masked coordinates according to the prescribed noise schedule. Newly revealed tokens are sampled from $\mu^{j,\eta}_{\theta,t_\ell}(\cdot \mid x)$, while previously revealed tokens remain fixed. We treat the fixed sampling temperature η as part of the finite-step sampling procedure and suppress it in the notation P_{θ,t_ℓ}. We write $\mathbb{P}_\theta$ for the resulting path law and p_θ for its terminal distribution at t_0.

3.2 Diffusion Guidance via Doob's h-Transform

Reward-guided diffusion reweights the terminal distribution of a pretrained model (Wu et al., 2023; Li et al., 2025). Following the reward-tilting perspective in KL-regularized reinforcement learning and stochastic control (Kappen, 2005; Levine, 2018), we parameterize the terminal weight by the reward function $R : \mathcal{X} \to \mathbb{R}$ of Sec. 1.

Definition 1 (Guided Terminal Distribution)**.** *Given a reward function $R : \mathcal{X} \to \mathbb{R}$ and a reward temperature $\beta > 0$, the corresponding guided terminal distribution is*

$$\pi_R(x_0) = \frac{p_\theta(x_0)\exp\left(R(x_0)/\beta\right)}{Z_R}, \qquad Z_R = \mathbb{E}_{X_0\sim p_\theta}\left[\exp\left(R(X_0)/\beta\right)\right]. \tag{5}$$

We refer to the problem of modifying the reverse dynamics of the pretrained diffusion model so that their terminal marginal is π_R as terminal-reweighted diffusion guidance.

Both soft objectives and hard constraints can be represented through the choice of R. A soft objective assigns a scalar reward $R(x) \in \mathbb{R}$ to each design, with larger values indicating greater preference. For a hard constraint specified by a feasible set $\mathcal{A} \subseteq \mathcal{X}$, we use the indicator-based reward function

$$R(x) = -P\mathbf{1}\{x \notin \mathcal{A}\}, \qquad P > 0. \tag{6}$$

In the limit $P \to \infty$, $\exp\left(R(x)/\beta\right)$ reduces to $\mathbf{1}\{x \in \mathcal{A}\}$, recovering exact conditioning on the feasible set. Thus, soft objectives and hard constraints share the same reward-tilting formulation.

Doob's h-transform provides an exact solution to the terminal-reweighting problem for Markov processes (Rogers & Williams, 2000; Särkkä & Solin, 2019). More generally, for a positive terminal weight $\phi : \mathcal{X} \to \mathbb{R}_{>0}$, define the corresponding terminal-reweighted distribution by

$$\pi_\phi(x_0) = \frac{p_\theta(x_0)\phi(x_0)}{\mathbb{E}_{X_0\sim p_\theta}[\phi(X_0)]}. \tag{7}$$

Definition 1 corresponds to the choice $\phi(x_0) = \exp(R(x_0)/\beta)$. To construct a reverse process with terminal distribution π_ϕ, we propagate the terminal weight ϕ backward through the original reverse dynamics using Doob's h-function.

Definition 2 (Doob's h-Function)**.** *For the terminal weight ϕ, define the backward value function*

$$h_\ell(x) = \mathbb{E}_{\mathbb{P}_\theta}\left[\phi(X_0) \mid X_{t_\ell} = x\right], \qquad h_0(x_0) = \phi(x_0). \tag{8}$$

The h-function propagates the terminal preference backward through the original reverse process; it is the soft value function of the corresponding control problem (Uehara et al., 2025b). Using this function, Doob's h-transform reweights the reverse transitions so as to realize the desired terminal distribution.

Proposition 1 (Doob's h-Transform)**.** *Let p_θ denote the terminal distribution at t_0 of the original reverse sampler initialized from $X_{t_L} = x_L$. Define the transformed reverse transition kernel by*

$$P^h_{\theta,t_\ell}(y \mid x) = P_{\theta,t_\ell}(y \mid x)\frac{h_{\ell-1}(y)}{h_\ell(x)}. \tag{9}$$

Suppose the reverse sampler is initialized from the same fixed initial state $X_{t_L} = x_L$ as the original diffusion sampler. Then P^h_{θ,t_ℓ} is a valid Markov transition kernel, and the terminal marginal of the transformed chain is exactly π_ϕ.

The proof of Proposition 1 is deferred to Appendix A. In masked discrete diffusion, the reverse sampler always starts from the same fully masked state $X_{t_L} = \mathtt{m}^N$. Hence, the fixed-initial-state condition of Proposition 1 is naturally satisfied, and Doob's h-transform changes the terminal marginal from p_θ to π_ϕ by reweighting the pretrained reverse transitions.

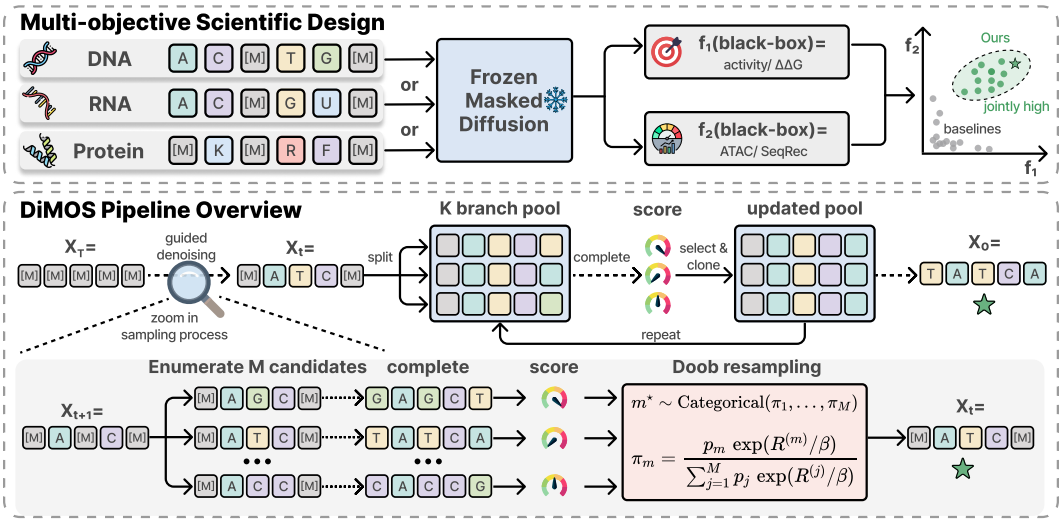


Figure 2: **Overview of DiMOS. Top:** multi-objective design with a frozen prior. **Middle:** K branches from step s, with reward-guided refinement. **Bottom:** M token candidates are enumerated, completed, and stochastically resampled using prior and reward weights.

4 Method

In this section, we introduce DiMOS for multi-objective scientific design using a frozen masked diffusion model. We first express the property requirements as a joint reward (Sec. 4.1). Using this feedback, approximate Doob-guided local resampling compares candidate updates and samples the next state without reward gradients (Sec. 4.2). We then introduce budget-efficient trajectory search to concentrate computation on promising continuations (Sec. 4.3).

4.1 Joint Reward for Multiple Requirements

In scientific design, candidates often need to meet multiple property thresholds. We therefore instantiate the feasible-set formulation (hard constraints) of Sec. 3.2 with per-property thresholds, rather than directly maximizing a weighted sum of per-property rewards (soft objectives), which does not in general guarantee joint feasibility. This setting also challenges gradient-based guidance: pass-or-fail constraints provide no useful gradient signal for correcting violations, and reward models may themselves be non-differentiable.

We consider n reward functions $R_k : \mathcal{X} \to \mathbb{R}$, each with an acceptance threshold c_k and a direction $\epsilon_k \in \{1, -1\}$, where 1 denotes a lower-bound requirement and -1 an upper-bound requirement. Define the signed constraint margin $d_k(x) = \epsilon_k(R_k(x) - c_k)$. Requirement k is satisfied if and only if $d_k(x) \geq 0$. The joint feasible set is therefore $\mathcal{A} = \{x : \ d_k(x) \geq 0 \text{ for all } k\}$. A single indicator of $\mathcal{A}$ assigns the same penalty to every infeasible design, regardless of which requirements are satisfied. To retain feedback about individual requirements, we encode each using a hard or soft gate and aggregate in log space,

$$R(x) = \sum_{k=1}^{n} w_k \, g_k\big(d_k(x)\big), \qquad g_k(d) = \begin{cases} -P\,\mathbf{1}\{d < 0\} & \text{(hard gate)}, \\ \log \sigma(d/\tau_k) & \text{(soft gate)}, \end{cases} \tag{10}$$

where σ is the logistic sigmoid, $w_k > 0$ weights each requirement, and $\tau_k > 0$ controls the soft gate's transition width. A hard gate returns 0 when the requirement is satisfied and $-P$ otherwise. A soft gate remains negative and increases continuously toward 0 as the margin grows, so it still distinguishes candidates above the threshold. Both gate types use the same feasibility criteria while providing different reward feedback. We retain the finite penalty $P > 0$ of Eq. (6), so violating a hard requirement does not force its weight to zero. Under the exponential tilt, log-space aggregation yields multiplicative per-requirement factors (Appendix D).

4.2 Approximate Doob-Guided Local Resampling

Doob's h-transform connects terminal reward to decisions during denoising. Using $\phi(x_0) = \exp(R(x_0)/\beta)$ with the joint reward in Eq. (10), Proposition 1 gives the relative weight of a candidate next state y from the current state x at time t_ℓ,

$$P^h_{\theta,t_\ell}(y \mid x) \propto P_{\theta,t_\ell}(y \mid x)\, h_{\ell-1}(y), \tag{11}$$

since the normalizer $h_\ell(x)$ is shared by all candidates. Guidance thus requires relative future values, but evaluating $h_{\ell-1}(y)$ entails averaging over all remaining continuations.

Completion-Based Value Estimation. To approximate $h_{\ell-1}(y)$, we complete the masked state y to a clean sequence $\hat{x}_0(y)$ using the frozen denoiser, and use its terminal weight as a proxy (Li et al., 2025; Uehara et al., 2025a):

$$\widehat{h}_{\ell-1}(y) = \phi\big(\hat{x}_0(y)\big) = \exp\big(R(\hat{x}_0(y))/\beta\big). \tag{12}$$

Enumeration and Stochastic Resampling. To avoid repeated token proposals, we enumerate distinct alternatives at one newly revealed position. At a guided step t_ℓ, fix $x = x_{t_\ell}$ and draw $Z \sim P_{\theta,t_\ell}(\cdot \mid x)$. Let $E = \{u : x^u = \mathtt{m},\, Z^u \in \mathcal{V}\}$ be the reveal set. When $E \neq \varnothing$, select the position $J \in E$ with the largest denoiser entropy, breaking ties by index, and enumerate the $M \le |\mathcal{V}|$ most probable tokens $a_1, \dots, a_M$ under $\mu^{J,\eta}_{\theta,t_\ell}(\cdot \mid x)$. For each candidate $y^{(m)} = Z^{J \leftarrow a_m}$, assign

$$\log \widehat{w}_m = \log \mu^{J,\eta}_{\theta,t_\ell}(a_m \mid x) + \log \widehat{h}_{\ell-1}(y^{(m)}). \tag{13}$$

We sample $m^\star \sim \mathrm{Categorical}(\mathrm{softmax}(\log \widehat{w}_{1:M}))$ and retain $y^{(m^\star)}$. Under the factorized token update, all other transition factors are shared; the token prior appears explicitly because alternatives are enumerated rather than sampled from it. Each update uses at most M reward evaluations without reward gradients.

We next bound the discrepancy from the full Doob transition, separating context-reweighting, top-M truncation, and value-estimation errors. For this analysis, assume $E \neq \varnothing$ almost surely.

The shared context $C = (E, (Z^u)_{u \in E \setminus \{J\}})$ records the reveal set and all newly revealed tokens except the one at J. Its distribution under the base transition is $\nu_\ell(c \mid x) = \mathbb{P}_{Z \sim P_{\theta,t_\ell}(\cdot|x)}(C = c)$. The context value $\Gamma_\ell(c; x) = \mathbb{E}[h_{\ell-1}(Z) \mid C = c]$ averages over the token at J under the same base transition. Below, expectations and variances indexed by ν_ℓ are over $C \sim \nu_\ell(\cdot \mid x)$.

Given $C = c$, the token at J has conditional law $p^h_\ell(\cdot \mid c, x)$ under the exact Doob transition $P^h_{\theta,t_\ell}(\cdot \mid x)$. The enumerated tokens form $S_M(c) = \{a_1, \dots, a_M\} \subseteq \mathcal{V}$. Restricting the exact conditional to this set and renormalizing gives

$$p^h_{\ell,M}(a \mid c, x) = \frac{p^h_\ell(a \mid c, x)\mathbf{1}\{a \in S_M(c)\}}{p^h_\ell(S_M(c) \mid c, x)}.$$

Using $\widehat{h}_{\ell-1}$ instead induces the practical conditional $\widehat{p}_{\ell,M}(\cdot \mid c, x)$, averaged over any completion randomness. Combining this conditional with the unchanged context law $\nu_\ell(\cdot \mid x)$ defines the full-state transition $\widehat{P}^{(M)}_{\theta,t_\ell}(\cdot \mid x)$.

Theorem 1 (Full-state error of coordinate-wise Doob resampling)**.** *For any $1 \le M \le |\mathcal{V}|$, suppose $\Gamma_\ell(C; x) > 0$ and $p^h_\ell(S_M(C) \mid C, x) > 0$ almost surely under $\nu_\ell(\cdot \mid x)$. Then $h_\ell(x) = \mathbb{E}_{\nu_\ell}[\Gamma_\ell(C; x)]$ and*

$$\begin{aligned}\mathrm{TV}\Big(\widehat{P}^{(M)}_{\theta,t_\ell}(\cdot \mid x), P^h_{\theta,t_\ell}(\cdot \mid x)\Big) \;\le\; & \underbrace{\frac{1}{2}\mathbb{E}_{\nu_\ell}\left[\left|\frac{\Gamma_\ell(C; x)}{h_\ell(x)} - 1\right|\right]}_{\textit{context-reweighting error}} + \underbrace{\mathbb{E}_{\nu_\ell}\big[p^h_\ell(S_M(C)^c \mid C, x)\big]}_{\textit{top-}M\textit{ truncation error}} \\ & + \underbrace{\mathbb{E}_{\nu_\ell}\big[\mathrm{TV}\big(\widehat{p}_{\ell,M}(\cdot \mid C, x), p^h_{\ell,M}(\cdot \mid C, x)\big)\big]}_{\textit{value-estimation error}}.\end{aligned} \tag{14}$$

The first term is at most $\sqrt{\mathrm{Var}_{\nu_\ell}[\Gamma_\ell(C; x)]}/(2h_\ell(x))$.

The proof and empirical diagnostics are in Appendices B and C. When $M = |\mathcal{V}|$, the truncation term vanishes. If the completion-based values approximate the exact Doob values well, the value-estimation term is also small, leaving the discrepancy primarily controlled by the variation of the shared-context value.

Table 1: **DNA design with hard and soft gates.** Activity + ATAC results with held-out activity evaluation. Both modes use the same success criteria, differing only in whether activity receives a hard gate (binary penalty) or a soft gate (smooth reward).

Method	Joint SR (%) ↑	Activity ↑		ATAC ↑		Uniq. (%)	3-mer r	JASPAR r	Time (ms)
		Median	Top-10	Median	Top-10				
Hard-Gate Mode									
Pretrained	0.00±0.00	0.18±0.01	2.41±0.39	0.03±0.00	0.65±0.06	100.00±0.00	-0.04±0.01	0.29±0.01	8.46±0.23
Best-of-N	2.34±0.34	1.37±0.12	5.32±0.19	0.45±0.01	0.90±0.04	100.00±0.00	0.76±0.00	0.90±0.00	271.56±0.68
TDS	78.80±6.63	4.80±0.13	6.19±0.21	0.84±0.02	0.97±0.01	48.12±1.29	0.92±0.03	0.91±0.01	171.39±0.18
SMC	76.04±8.17	4.74±0.11	6.25±0.24	0.83±0.03	0.97±0.01	48.02±2.62	0.92±0.04	0.91±0.01	89.28±0.10
FK-Steering	4.90±0.58	0.84±0.06	5.51±0.14	0.30±0.01	0.92±0.01	100.00±0.00	0.62±0.01	0.86±0.00	111.00±0.83
SVDD	45.05±0.97	4.14±0.03	6.62±0.04	0.89±0.01	0.99±0.00	100.00±0.00	0.69±0.00	0.91±0.00	244.61±0.53
DiMOS (Ours)	**87.55**±0.65	**5.34**±0.05	**6.98**±0.07	**0.99**±0.00	**1.00**±0.00	100.00±0.00	0.86±0.00	0.93±0.00	234.63±0.33
Soft-Gate Mode									
Pretrained	0.00±0.00	0.18±0.01	2.41±0.39	0.03±0.00	0.65±0.06	100.00±0.00	-0.04±0.01	0.29±0.01	8.46±0.23
Best-of-N	2.08±0.39	1.88±0.03	5.29±0.22	0.42±0.01	0.90±0.03	100.00±0.00	0.79±0.01	0.91±0.01	240.96±0.65
TDS	47.40±7.16	4.14±0.15	6.15±0.33	0.79±0.02	0.95±0.01	47.97±1.55	0.79±0.02	0.88±0.02	200.87±0.93
SMC	41.20±5.15	4.02±0.18	5.71±0.05	0.81±0.02	0.95±0.02	48.39±1.48	0.74±0.05	0.86±0.01	89.73±0.15
FK-Steering	4.84±0.34	1.45±0.07	5.76±0.12	0.25±0.00	0.92±0.01	100.00±0.00	0.61±0.02	0.85±0.01	110.62±0.32
SVDD	55.16±1.55	4.66±0.01	7.24±0.09	0.83±0.01	**0.99**±0.00	100.00±0.00	0.74±0.01	0.93±0.00	244.70±0.56
DiMOS (Ours)	**98.39**±0.29	**5.95**±0.02	**7.32**±0.06	**0.99**±0.00	**0.99**±0.00	100.00±0.00	0.88±0.00	0.93±0.00	234.60±0.57

4.3 Budget-Efficient Trajectory Search

Jointly satisfying multiple requirements can leave a small feasible region (Xia et al., 2025), making repeated independent sampling costly. We therefore share trajectory prefixes and allocate more search budget to branches with higher estimated joint rewards.

Stage-Wise Branching. To control the cost of multi-trajectory exploration, we share a denoising prefix within each of B independent generation groups, each returning one design. Each group follows a single trajectory for s steps, then copies its partially denoised state into K branches. These branches explore alternative continuations through the stochastic local updates of Sec. 4.2. The branch width K determines how many continuations are explored concurrently, while the split step s trades a longer shared prefix for a shorter branching phase.

Reward-Guided Refinement. At post-split checkpoints, we score branches using the completion-based estimate of Sec. 4.2. Within each group, we copy the state with the highest estimated joint reward into all K slots, then use stochastic local updates to explore new continuations. Each group returns its highest-reward final design. Group-local selection prevents one trajectory from replacing other groups' outputs, helping preserve sequence uniqueness. Prefix sharing reduces state updates over L steps from KL to $s + K(L - s)$ per group. Each guided update compares up to M candidates per branch, with additional reward evaluations for branch selection (Appendix E).

5 Experiments

In this section, we evaluate DiMOS on DNA, protein, and RNA design using frozen generative priors. We assess joint success rate (SR) and design quality under hard-gate and soft-gate modes, then examine component contributions, hyperparameter sensitivity, and time-budget trade-offs.

5.1 Experimental Setup

Tasks and Gate Modes. We study ❶ **Activity + ATAC** for DNA regulatory activity and chromatin accessibility (Gosai et al., 2024; Wang et al., 2025), and ❷ $\Delta\Delta G$ **+ SeqRec** for protein stability and sequence recovery. Both tasks require joint threshold satisfaction. In hard-gate mode, activity or $\Delta\Delta G$ uses the binary penalty of Eq. (10), while the other objective retains its soft gate. In soft-gate mode, both objectives use soft gates. Success criteria remain unchanged. This tests guidance when one requirement provides no useful gradient. Additional DNA, protein, and RNA results appear in Appendix I.

Table 2: **Protein design with hard and soft gates.** $\Delta\Delta G$ + SeqRec results with identical success criteria. The $\Delta\Delta G$ term uses a hard gate (binary penalty) or a soft gate (smooth reward).

Method	Joint SR (%) ↑	$\Delta\Delta G$ ↑		SeqRec ↑		Uniq. (%)	scRMSD (%)	pLDDT (%)	Time (ms)
		Median	Top-10	Median	Top-10				
Hard-Gate Mode									
Pretrained	4.04±0.30	-0.45±0.00	0.45±0.03	0.44±0.00	0.48±0.00	99.67±0.05	93.55±0.11	87.98±0.11	17.42±0.08
Best-of-N	16.06±0.19	-0.02±0.01	0.70±0.05	0.49±0.00	0.52±0.00	99.18±0.03	92.84±0.09	86.65±0.16	159.89±0.05
TDS	13.59±0.48	-0.12±0.01	0.38±0.06	0.47±0.00	0.50±0.00	16.75±0.24	93.12±1.00	87.96±0.21	124.49±0.38
SMC	4.10±0.30	-0.10±0.06	0.69±0.07	0.44±0.00	0.48±0.00	22.09±0.74	92.90±0.83	89.37±0.03	83.11±0.08
FK-Steering	8.03±0.08	-0.26±0.01	0.71±0.03	0.45±0.00	0.50±0.00	99.93±0.00	94.23±0.16	87.43±0.19	192.43±0.12
SVDD	31.68±0.03	**0.07**±0.01	0.75±0.02	0.54±0.00	0.57±0.00	98.61±0.12	93.86±0.13	84.64±0.09	166.79±0.09
DiMOS (Ours)	**58.38**±0.36	**0.07**±0.00	**1.38**±0.00	**0.61**±0.00	**0.63**±0.00	100.00±0.00	92.32±0.05	85.92±0.03	157.63±0.05
Soft-Gate Mode									
Pretrained	4.04±0.30	-0.45±0.00	0.45±0.03	0.44±0.00	0.48±0.00	99.67±0.05	93.55±0.11	87.98±0.11	17.38±0.10
Best-of-N	17.77±0.19	0.01±0.01	0.73±0.04	0.49±0.00	0.52±0.00	99.00±0.08	94.21±0.05	85.81±0.05	159.68±0.13
TDS	9.14±0.45	-0.31±0.02	0.41±0.04	0.47±0.00	0.49±0.00	17.34±0.19	94.88±0.59	90.67±0.17	123.78±0.58
SMC	6.79±0.43	-0.48±0.05	0.53±0.03	0.45±0.00	0.48±0.00	22.14±0.97	92.25±0.37	86.24±0.21	82.51±0.03
FK-Steering	6.38±0.19	-0.33±0.00	0.44±0.01	0.46±0.00	0.50±0.00	99.54±0.00	93.84±0.11	87.63±0.11	192.32±0.06
SVDD	35.66±0.41	0.38±0.01	1.00±0.02	0.53±0.00	0.56±0.00	96.83±0.21	96.57±0.12	83.90±0.03	166.70±0.08
DiMOS (Ours)	**70.64**±0.24	**0.51**±0.01	**1.39**±0.03	**0.60**±0.00	**0.63**±0.00	99.93±0.00	93.99±0.21	85.61±0.09	157.51±0.11

Baselines. We compare sampling and selection (PRETRAINED, BEST-OF-N), gradient-based steering (TDS) (Wu et al., 2023; Nisonoff et al., 2025), and gradient-free steering (SMC, FK-STEERING, SVDD) (Singhal et al., 2025; Li et al., 2025). All methods share the frozen prior within each task, and most baselines have generation times of the same order of magnitude. Implementation and budget details are in Appendix G.

Evaluation Metrics. We report joint SR, the fraction of outputs satisfying both thresholds, and each objective's median over all outputs and over the ten designs with the highest joint reward. We also report uniqueness (the fraction of distinct sequences), domain-specific naturalness and validity checks, and generation time per design. Tabs. 1 and 2 report mean ± s.d. over three seeds. Full metric definitions and thresholds are in Appendix F.

5.2 MAIN RESULTS

Tabs. 1 and 2 summarize DNA and protein results under both gate modes. Three findings emerge. ❶ **Joint Objective Improvement.** DiMOS leads both domains and gate modes in joint SR, including with a non-differentiable hard gate. Its evaluation-only guidance uses reward values without smoothing that gate. In soft-gate mode, it nearly doubles joint SR relative to SVDD, the strongest baseline, while raising both objective medians, showing distribution-wide gains beyond top-ranked designs. ❷ **Effective Generation-Time Guidance.** At comparable generation times, DiMOS substantially outperforms BEST-OF-N, which ranks only completed samples. This supports integrating reward feedback into generation rather than relying solely on final selection to find jointly feasible designs within a given budget. ❸ **Diverse High-Quality Outputs.** DiMOS retains 100% uniqueness on DNA and over 99.9% on protein under both gate modes. In DNA hard-gate mode, distinct successful designs account for 87.55% of its outputs, versus at most 48.12% for TDS, bounded by its uniqueness. This is consistent with restricting branch replication to independent generation groups (Sec. 4.3). DiMOS retains high DNA motif-profile correlations and over 92% protein fold recovery (scRMSD), providing quality checks beyond the optimized objectives.

5.3 HYPERPARAMETER ANALYSIS AND ABLATION

Hyperparameter Sensitivity. Fig. 3 examines candidate width M, branch width K, and normalized search onset s/T on DNA and protein. Increasing M and K generally improves joint SR in both gate modes, although gains diminish as more branches are added. Search onset trades prefix sharing against multi-trajectory exploration. DNA maintains high success when branching is delayed through roughly the first half of denoising. A later split propagates one shared trajectory instead of K branches for more steps, allowing DNA to save computation on the prefix with little loss in success. Protein benefits from earlier branching, favoring more computation on alternative continuations rather than a long shared prefix. The allocation should therefore reflect the task.

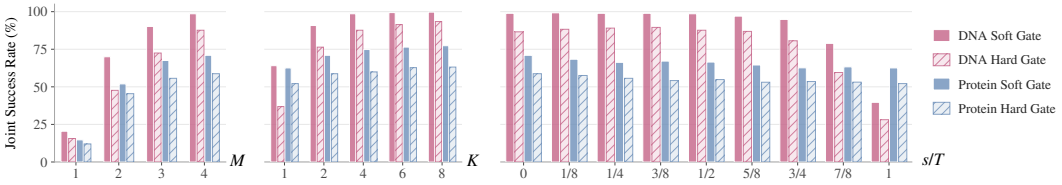


Figure 3: **Sensitivity to M, K, and s/T.** One parameter varies at a time.

Table 3: **Component ablation.** Each row removes one component; entries are the change in joint SR (%) from the full configuration, over three seeds.

Configuration	DNA		Protein	
	Hard Gate	Soft Gate	Hard Gate	Soft Gate
Full DiMOS	87.55±0.65	98.39±0.29	58.38±0.36	70.64±0.24
Doob-guided local resampling				
– Doob guidance	-62.92±1.84	-64.95±0.46	-46.96±0.24	-56.18±0.35
– enumeration	-12.66±1.72	-5.26±0.22	-38.28±0.08	-47.61±0.08
Budget-efficient trajectory search				
– stage-wise branching	-51.61±2.03	-34.69±3.29	-6.47±0.19	-7.75±0.23
– reward-guided refinement	-18.39±1.63	-7.03±1.10	-0.11±0.27	-1.80±0.08

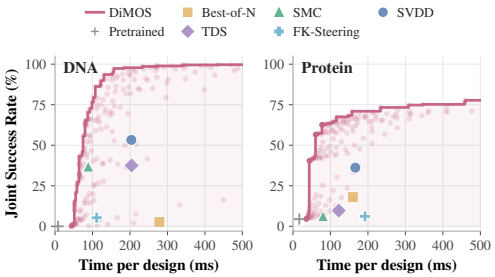


Figure 4: **Joint success versus generation time.** DiMOS budget envelopes on Activity + ATAC (DNA) and $\Delta\Delta G$ + SeqRec (protein).

Component Contributions. Tab. 3 tests what each component adds beyond the remaining algorithm. ❶ **Local Reward Feedback.** Removing Doob guidance disables the reward term in local resampling but retains candidate priors and reward-based branch selection. This causes the largest drop in joint SR, showing that trajectory-level selection alone cannot replace reward-informed token updates. ❷ **Candidate Construction.** Replacing enumeration with M i.i.d. token draws at the same selected position allows repeats while keeping the candidate count and reweighting rule unchanged. The larger losses on protein highlight the value of evaluating distinct alternatives, not merely more candidates. ❸ **Multiple Continuations.** Setting $K = 1$ retains local guidance but collapses search to a single trajectory. The larger losses on DNA highlight the value of exploring multiple continuations rather than committing to one guided path. ❹ **Intermediate Selection.** Disabling refinement retains the K branches and final best-of-K selection but removes intermediate pruning and replication. The losses, especially on DNA, show that multiple trajectories benefit from redirecting computation toward promising partial designs during search. App. H transplants this search layer onto the SVDD kernel at matched wall clock: it improves that kernel too, but closes under a fifth of the gap to DiMOS, leaving the local transition as the larger factor.

5.4 Efficiency under Time Budgets

Fig. 4 plots an empirical Pareto frontier between generation time and joint SR from single-seed, soft-gate sweeps. The staircase gives the highest observed joint SR within each time budget, revealing two benefits. ❶ **Higher Success at a Given Budget.** At the time budgets of the compared reward-based baselines, DiMOS achieves higher joint SR in both domains. It therefore uses the available generation budget to produce a higher proportion of jointly feasible designs. ❷ **Lower Cost at a Given Success Rate.** Measured DiMOS configurations match or exceed SVDD's joint SR with approximately 63% less generation time on DNA and 74% less on protein. Practitioners can therefore choose higher success within a fixed budget or lower cost for a target success rate.

6 Conclusion

We introduced DiMOS, a training-free framework for multi-objective scientific design with frozen masked diffusion models. It combines approximate Doob-guided local resampling with budget-efficient trajectory search, accommodating hard constraints and black-box rewards without gradients. Experiments across DNA, protein, and RNA show improved joint success while retaining diverse designs. Allocation studies further highlight task-dependent trade-offs between candidate width, branching, and search duration.

AI USE STATEMENT

We used large language models (ChatGPT/Claude/Gemini) for grammatical correction in this manuscript. LLMs played no role in research ideation or methodology. All technical contributions, analyses, and scientific conclusions are the authors' own.

ETHICS STATEMENT

DiMOS is evaluated computationally on DNA, protein, and RNA design. No generated sequences were experimentally validated in this study. Improvements in computational rewards and sequence or structural checks do not establish biological efficacy or safety. Biases in the frozen priors and reward models, and optimization beyond their reliable domains, may produce misleading candidates. Because sequence design can alter biological activity, downstream synthesis and use require independent validation and appropriate institutional oversight and biosafety review.

REPRODUCIBILITY STATEMENT

Sec. 4 specifies the method and its approximations, and Sec. 5 describes the experimental comparisons. Proofs are provided in Appendices A and B. Appendix E gives sampler pseudocode, schedules, and configurations. Appendix F defines success thresholds and metrics, while Appendix G documents baseline and budget protocols, output counts, and reward-model separation between guidance and evaluation. Additional tasks are reported in Appendix I. Main comparisons report mean $\pm$ s.d. over three seeds, while the time-budget sweeps use a single seed. Code repository: `https://anonymous.4open.science/r/DiMOS/`.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html`.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html`.

Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M. Jude, Iva Marković, Rameshwar U. Kadam, Koen H. G. Verschueren, Kenneth Verstraete, Scott Thomas Russell Walsh, Nathaniel Bennett, Ashish Phal, Aerin Yang, Lisa Kozodoy, Michelle DeWitt, Lora Picton, Lauren Miller, Eva-Maria Strauch, Nicholas D. DeBouver, Allison Pires, Asim K. Bera, Samer Halabiya, Bradley Hammerson, Wei Yang, Steffen Bernard, Lance Stewart, Ian A. Wilson, Hannele Ruohola-Baker, Joseph Schlessinger, Sangwon Lee, Savvas N. Savvides, K. Christopher Garcia, and David Baker. Design of protein-binding proteins from the target structure alone. *Nature*, 605:551–560, 2022. doi: 10.1038/s41586-022-04654-9. URL `https://www.nature.com/articles/s41586-022-04654-9`.

Jaime A. Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, Oriol Fornes, Tiffany Y. Leung, Alejandro Aguirre, Fayrouz Hammal, Daniel Schmelter, Damir Baranasic, Benoit Ballester, Albin Sandelin, Boris Lenhard, Klaas Vandepoele, Wyeth W. Wasserman, François Parcy, and Anthony Mathelier. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 50(D1): D165–D173, 2022. doi: 10.1093/nar/gkab1113. URL `https://academic.oup.com/nar/article/50/D1/D165/6446529`.

Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow matching for controllable biological sequence design. In *ICML Workshop on Multi-modal*

Foundation Models and Large Language Models for Life Sciences, 2025a. URL https://arxiv.org/abs/2505.07086.

Xingyu Chen, Shihao Ma, Runsheng Lin, Jiecong Lin, and Bo Wang. Ctrl-DNA: Controllable cell-type-specific regulatory DNA design via constrained RL. In *Advances in Neural Information Processing Systems*, 2025b. URL https://arxiv.org/abs/2505.20578.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/13e8be77982beb73d7ed0bbf122f9f3c-Abstract-Conference.html.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01781.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/852f50969a9e523ec41d26f2f68bd456-Abstract-Conference.html.

Sager J. Gosai, Rodrigo I. Castro, Natalia Fuentes, John C. Butts, Kousuke Mouri, Michael Alasoadura, Susan Kales, Thanh Thanh L. Nguyen, Ramil R. Noche, Arya S. Rao, Mary T. Joy, Pardis C. Sabeti, Steven K. Reilly, and Ryan Tewhey. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, 634:1211–1220, 2024. doi: 10.1038/s41586-024-08070-z. URL https://www.nature.com/articles/s41586-024-08070-z.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12489–12517, 2023. doi: 10.52202/075280-0547. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/29591f355702c3f4436991335784b503-Paper-Conference.pdf.

Wei Guo, Yuchen Zhu, Molei Tao, and Yongxin Chen. Plug-and-play controllable generation for discrete masked models. *arXiv preprint arXiv:2410.02143*, 2024. URL https://arxiv.org/abs/2410.02143.

Jiaqi Han, Austin Wang, Minkai Xu, Wenda Chu, Meihua Dang, Haotian Ye, Huayu Chen, Yisong Yue, and Stefano Ermon. Discrete diffusion trajectory alignment via stepwise decomposition. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0922ceebb39f1c76e64ef03d0885fa82-Abstract-Conference.html.

Hilbert J Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201, 2005.

Jack Kyte and Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982. doi: 10.1016/0022-2836(82)90515-0.

Kathrin Leppek, Gun Woo Byeon, Wipapat Kladwang, Hannah K. Wayment-Steele, Craig H. Kerr, Adele F. Xu, Do Soon Kim, Ved V. Topkar, Christian Choe, Daphna Rothschild, Gerald C. Tiu, Roger Wellington-Oguri, Kotaro Fujii, Eesha Sharma, Andrew M. Watkins, John J. Nicol, Jonathan Romano, Bojan Tunguz, Fernando Diaz, Hui Cai, Pengbo Guo, Jiewei Wu, Fanyu Meng, Shuai Shi, Eterna Participants, Philip R. Dormitzer, Alicia Solórzano, Maria Barna, and Rhiju Das. Combinatorial optimization of mRNA structure, stability, and translation for RNA-based therapeutics. *Nature Communications*, 13:1536, 2022. doi: 10.1038/s41467-022-28776-w. URL https://www.nature.com/articles/s41467-022-28776-w.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 95507–95545, 2025. doi: 10.52202/085713-3194. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/899af0d66d8850318a20781484416152-Paper-Conference.pdf.

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. URL https://www.science.org/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.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Toan Nguyen, Kien Do, Duc Kieu, and Thin Nguyen. h-Edit: Effective and flexible diffusion-based editing via Doob's h-transform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28490–28501, 2025.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/597254dc45be8c166d3ccf0ba2d56325-Abstract-Conference.html.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Zachary Quinn, Chenghao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Yoshua Bengio, Pranam Chatterjee, Alexander Tong, and Avishek Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3f2a9f41dd0cb07cd9b60c058c7b1682-Paper-Conference.pdf.

L. Chris G. Rogers and David Williams. *Diffusions, Markov Processes, and Martingales*, volume 2. Cambridge University Press, 2000.

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, pp. 130136–130184, 2024. doi: 10.52202/079017-4135. URL https://papers.nips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Paul J. Sample, Ban Wang, David W. Reid, Vlad Presnyak, Iain J. McFadyen, David R. Morris, and Georg Seelig. Human 5' UTR design and variant effect prediction from a massively parallel translation assay. *Nature Biotechnology*, 37(7):803–809, 2019.

Anirban Sarkar, Ziqi Tang, Chris Zhao, and Peter K. Koo. Designing DNA with tunable regulatory activity using discrete diffusion. *bioRxiv*, 2024. doi: 10.1101/2024.05.23.595630. URL https://www.biorxiv.org/content/10.1101/2024.05.23.595630v1.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander M. Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.10193.

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*, 2024. URL https://arxiv.org/abs/2406.04329.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *International Conference on Machine Learning*, volume 267, pp. 55810–55827, 2025. URL https://proceedings.mlr.press/v267/singhal25b.html.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20459–20478. PMLR, 2022. URL https://proceedings.mlr.press/v162/stanton22a.html.

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

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60515–60529. PMLR, 2025a. URL https://proceedings.mlr.press/v267/uehara25a.html.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025b. URL https://arxiv.org/abs/2501.09685.

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.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/63e8bc7bbf1cfea36d1d1b6538aecce5-Abstract.html.

Xin Xia, Yajie Zhang, Xiangxiang Zeng, Xingyi Zhang, Chunhou Zheng, and Yansen Su. CMOMO: a deep multi-objective optimization framework for constrained molecular multi-property optimization. *Briefings in Bioinformatics*, 26(4):bbaf335, 2025. doi: 10.1093/bib/bbaf335. URL https://academic.oup.com/bib/article/26/4/bbaf335/8195924.

Qijie Zhu, Zeqi Ye, Han Liu, Zhaoran Wang, and Minshuo Chen. Training-free adaptation of diffusion models via Doob's h-transform. *arXiv preprint arXiv:2602.16198*, 2026. URL https://arxiv.org/abs/2602.16198.

Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K. Speicher, Christoph S. Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, Verena Siewers, Florian David, Jens Nielsen, and Aleksej Zelezniak. Controlling gene expression with deep generative design of regulatory DNA. *Nature Communications*, 13:5099, 2022. doi: 10.1038/s41467-022-32818-8. URL https://www.nature.com/articles/s41467-022-32818-8.

A Proof of Proposition 1

Proof. By the Markov property of the original reverse process and the definition of h_ℓ, for any state x at time t_ℓ,

$$\begin{aligned} h_\ell(x) &= \mathbb{E}_{\mathbb{P}_\theta}\left[\phi(X_0) \mid X_{t_\ell} = x\right] \\ &= \mathbb{E}_{\mathbb{P}_\theta}\left[h_{\ell-1}(X_{t_{\ell-1}}) \mid X_{t_\ell} = x\right] \\ &= \sum_y P_{\theta,t_\ell}(y \mid x) h_{\ell-1}(y). \end{aligned} \tag{15}$$

Since ϕ is strictly positive, so is h_ℓ on the states reached by the reverse sampler. Therefore,

$$\sum_y P^h_{\theta,t_\ell}(y \mid x) = \frac{1}{h_\ell(x)} \sum_y P_{\theta,t_\ell}(y \mid x) h_{\ell-1}(y) = 1. \tag{16}$$

Moreover, $P^h_{\theta,t_\ell}(y \mid x) \geq 0$, so $P^h_{\theta,t_\ell}(\cdot \mid x)$ is a valid transition kernel.

Let $\mathbb{P}^h_\theta$ denote the path law induced by the transformed kernels. For a trajectory $x_L, x_{L-1}, \ldots, x_0$ starting from the fixed state x_L,

$$\begin{aligned} &\mathbb{P}^h_\theta\left(X_{t_{L-1}} = x_{L-1}, \ldots, X_{t_0} = x_0 \mid X_{t_L} = x_L\right) \\ &\quad = \prod_{\ell=1}^{L} P^h_{\theta,t_\ell}(x_{\ell-1} \mid x_\ell) \\ &\quad = \prod_{\ell=1}^{L} P_{\theta,t_\ell}(x_{\ell-1} \mid x_\ell) \frac{h_{\ell-1}(x_{\ell-1})}{h_\ell(x_\ell)} \\ &\quad = \left[\prod_{\ell=1}^{L} P_{\theta,t_\ell}(x_{\ell-1} \mid x_\ell)\right] \frac{\phi(x_0)}{h_L(x_L)}, \end{aligned} \tag{17}$$

where the h-ratios telescope and $h_0(x_0) = \phi(x_0)$.

Summing over $x_1, \ldots, x_{L-1}$ gives

$$\begin{aligned} \mathbb{P}^h_\theta\left(X_{t_0} = x_0 \mid X_{t_L} = x_L\right) &= \frac{\phi(x_0)}{h_L(x_L)} \mathbb{P}_\theta\left(X_{t_0} = x_0 \mid X_{t_L} = x_L\right) \\ &= \frac{p_\theta(x_0)\phi(x_0)}{h_L(x_L)}. \end{aligned} \tag{18}$$

By definition of the h-function and because the original sampler starts from the fixed state x_L,

$$\begin{aligned} h_L(x_L) &= \mathbb{E}_{\mathbb{P}_\theta}\left[\phi(X_0) \mid X_{t_L} = x_L\right] \\ &= \sum_{x_0} p_\theta(x_0)\phi(x_0) = \mathbb{E}_{X_0 \sim p_\theta}[\phi(X_0)]. \end{aligned} \tag{19}$$

Substituting this into Eq. (18) yields

$$\mathbb{P}^h_\theta\left(X_{t_0} = x_0 \mid X_{t_L} = x_L\right) = \frac{p_\theta(x_0)\phi(x_0)}{\mathbb{E}_{X_0 \sim p_\theta}[\phi(X_0)]} = \pi_\phi(x_0).$$

Hence the transformed chain has terminal marginal π_ϕ. □

B Proof of Theorem 1

Proof. Fix $x = x_{t_\ell}$ and suppress the dependence on x and t_ℓ when there is no ambiguity. Let $A = Z^J$ denote the token at the selected coordinate J. For a context $c = (e, b)$ and token $a \in \mathcal{V}$, define $y_x(c, a)$ as the unique successor state whose reveal set is e, whose token at J is a, and whose other newly revealed tokens are given by b; all remaining coordinates agree with x. Define the corresponding base conditional by

$$p_\ell(a \mid c, x) = \mathbb{P}(A = a \mid C = c, X_{t_\ell} = x).$$

For fixed x, the correspondence between the successor Z and the pair (C, A) is one-to-one: Z determines its reveal set E, the selected coordinate J, and hence (C, A), while (C, A) reconstructs Z as $y_x(C, A)$. Therefore,

$$P_{\theta,t_\ell}(y_x(c, a) \mid x) = \nu_\ell(c \mid x)\, p_\ell(a \mid c, x). \tag{20}$$

By the definition of the context value,

$$\Gamma_\ell(c; x) = \sum_{a \in \mathcal{V}} p_\ell(a \mid c, x)\, h_{\ell-1}(y_x(c, a))\,.$$

Therefore, using the backward recursion for the Doob value,

$$\begin{aligned} h_\ell(x) &= \sum_y P_{\theta,t_\ell}(y \mid x)\, h_{\ell-1}(y) \\ &= \sum_c \nu_\ell(c \mid x) \sum_{a \in \mathcal{V}} p_\ell(a \mid c, x)\, h_{\ell-1}(y_x(c, a)) \\ &= \sum_c \nu_\ell(c \mid x)\, \Gamma_\ell(c; x) = \mathbb{E}_{\nu_\ell}\big[\Gamma_\ell(C; x)\big]. \end{aligned} \tag{21}$$

We next characterize the exact Doob transition. From Eq. (20),

$$\begin{aligned} P^h_{\theta,t_\ell}(y_x(c, a) \mid x) &= P_{\theta,t_\ell}(y_x(c, a) \mid x)\, \frac{h_{\ell-1}(y_x(c, a))}{h_\ell(x)} \\ &= \nu_\ell(c \mid x) \frac{\Gamma_\ell(c; x)}{h_\ell(x)} p^h_\ell(a \mid c, x), \end{aligned} \tag{22}$$

where

$$p^h_\ell(a \mid c, x) = \frac{p_\ell(a \mid c, x) h_{\ell-1}(y_x(c, a))}{\Gamma_\ell(c; x)}. \tag{23}$$

Thus the exact Doob transition changes both the conditional law of the token at J and the marginal law of the shared context. In particular, the latter becomes

$$\nu^h_\ell(c \mid x) = \nu_\ell(c \mid x) \frac{\Gamma_\ell(c; x)}{h_\ell(x)}. \tag{24}$$

To separate the three approximation sources, introduce two intermediate coordinate-wise kernels. First, let $\widetilde{P}^h_{\theta,t_\ell}$ keep $C \sim \nu_\ell(\cdot \mid x)$ unchanged and sample the token at J from the exact conditional p^h_ℓ:

$$\widetilde{P}^h_{\theta,t_\ell}(y_x(c, a) \mid x) = \nu_\ell(c \mid x)\, p^h_\ell(a \mid c, x). \tag{25}$$

Second, let $\widetilde{P}^{h,(M)}_{\theta,t_\ell}$ additionally restrict the token to the enumerated set $S_M(c)$:

$$\widetilde{P}^{h,(M)}_{\theta,t_\ell}(y_x(c, a) \mid x) = \nu_\ell(c \mid x)\, p^h_{\ell,M}(a \mid c, x). \tag{26}$$

Finally, by definition of the practical resampling rule,

$$\widehat{P}^{(M)}_{\theta,t_\ell}(y_x(c, a) \mid x) = \nu_\ell(c \mid x)\, \widehat{p}_{\ell,M}(a \mid c, x), \tag{27}$$

where $\widehat{p}_{\ell,M}$ includes the completion-based value approximation, averaged over any completion randomness.

We now bound the discrepancy between consecutive kernels.

Context-Reweighting Error. The kernels $\widetilde{P}^h_{\theta,t_\ell}$ and P^h_{θ,t_ℓ} have the same conditional law $p^h_\ell(\cdot \mid c, x)$ given $C = c$ and differ only in their context marginals. Using Eqs. (22) and (25),

$$\begin{aligned}
&\mathrm{TV}\Big(\widetilde{P}^h_{\theta,t_\ell}(\cdot \mid x), P^h_{\theta,t_\ell}(\cdot \mid x)\Big) \\
&\quad = \frac{1}{2}\sum_{c,a} \nu_\ell(c \mid x)\left|1 - \frac{\Gamma_\ell(c;x)}{h_\ell(x)}\right| p^h_\ell(a \mid c, x) \\
&\quad = \frac{1}{2}\mathbb{E}_{\nu_\ell}\left[\left|\frac{\Gamma_\ell(C;x)}{h_\ell(x)} - 1\right|\right].
\end{aligned} \tag{28}$$

Top-M Truncation Error. For each context c, $p^h_{\ell,M}$ is obtained by restricting p^h_ℓ to $S_M(c)$ and renormalizing. Hence

$$\mathrm{TV}\big(p^h_{\ell,M}(\cdot \mid c, x), p^h_\ell(\cdot \mid c, x)\big) = p^h_\ell(S_M(c)^c \mid c, x). \tag{29}$$

Indeed, writing $q_c = p^h_\ell(S_M(c) \mid c, x)$,

$$\begin{aligned}
&\frac{1}{2}\sum_{a\in\mathcal{V}} \left|p^h_{\ell,M}(a \mid c, x) - p^h_\ell(a \mid c, x)\right| \\
&\quad = \frac{1}{2}\left[\sum_{a\in S_M(c)} p^h_\ell(a \mid c, x)\left(\frac{1}{q_c} - 1\right) + \sum_{a\notin S_M(c)} p^h_\ell(a \mid c, x)\right] \\
&\quad = \frac{1}{2}\big[(1 - q_c) + (1 - q_c)\big] = 1 - q_c.
\end{aligned}$$

Since the two full-state kernels share the same context law ν_ℓ, Eq. (29) gives

$$\begin{aligned}
&\mathrm{TV}\Big(\widetilde{P}^{h,(M)}_{\theta,t_\ell}(\cdot \mid x), \widetilde{P}^h_{\theta,t_\ell}(\cdot \mid x)\Big) \\
&\qquad = \mathbb{E}_{\nu_\ell}\big[p^h_\ell(S_M(C)^c \mid C, x)\big].
\end{aligned} \tag{30}$$

Value-Estimation Error. The practical and truncated-oracle kernels also share the same context law ν_ℓ. Therefore,

$$\begin{aligned}
&\mathrm{TV}\Big(\widehat{P}^{(M)}_{\theta,t_\ell}(\cdot \mid x), \widetilde{P}^{h,(M)}_{\theta,t_\ell}(\cdot \mid x)\Big) \\
&\quad = \frac{1}{2}\sum_{c} \nu_\ell(c \mid x)\sum_{a\in\mathcal{V}} \left|\widehat{p}_{\ell,M}(a \mid c, x) - p^h_{\ell,M}(a \mid c, x)\right| \\
&\quad = \mathbb{E}_{\nu_\ell}\big[\mathrm{TV}\big(\widehat{p}_{\ell,M}(\cdot \mid C, x), p^h_{\ell,M}(\cdot \mid C, x)\big)\big].
\end{aligned} \tag{31}$$

Applying the triangle inequality along

$$\widehat{P}^{(M)}_{\theta,t_\ell} \;\longrightarrow\; \widetilde{P}^{h,(M)}_{\theta,t_\ell} \;\longrightarrow\; \widetilde{P}^h_{\theta,t_\ell} \;\longrightarrow\; P^h_{\theta,t_\ell}$$

and substituting Eqs. (28), (30), and (31) yields Eq. (14).

Finally, Eq. (21) implies $\mathbb{E}_{\nu_\ell}[\Gamma_\ell(C;x)] = h_\ell(x)$. By Cauchy–Schwarz,

$$\begin{aligned}
\frac{1}{2}\mathbb{E}_{\nu_\ell}\left[\left|\frac{\Gamma_\ell(C;x)}{h_\ell(x)} - 1\right|\right] &= \frac{\mathbb{E}_{\nu_\ell}[|\Gamma_\ell(C;x) - h_\ell(x)|]}{2h_\ell(x)} \\
&\le \frac{\sqrt{\mathrm{Var}_{\nu_\ell}[\Gamma_\ell(C;x)]}}{2h_\ell(x)}.
\end{aligned} \tag{32}$$

This proves the claimed bound. □

The branch-level pruning of Sec. 4.3 sits outside this bound: it serves search rather than exact sampling from the terminal-reweighted distribution.

C Empirical Evaluation of Local Approximation Error

We evaluate the single-step approximation in Theorem 1 on DNA Activity + ATAC, protein $\Delta\Delta G$ + sequence recovery, and RNA MRL + accessibility. For each domain and gate mode, we collect 24 partially generated sequences from three seeds and eight generation groups per seed. Each group contributes its first branch when three MASK positions remain, using the main-table configurations. Protein samples are balanced across the first six held-out backbones reaching this count.

This choice makes the reference calculation tractable. Exact h-values average terminal exponential rewards over all remaining base-process continuations, whose number grows rapidly with the number of MASK positions. Earlier in decoding, Monte Carlo estimation is more affordable but introduces sampling uncertainty, especially when a few continuations dominate the reward weights. Using three MASK positions permits exact enumeration across all three domains, isolating the sampler's approximation error from Monte Carlo reference error at a common near-terminal setting.

We compute the reference Doob transition by enumerating the remaining base-process continuations and compare it with the practical local resampling kernel. The reference includes all token classes, while practical menus and completion rules remain unchanged. Completion randomness is integrated where applicable, so the reported TV measures the full single-step discrepancy rather than only the context term. Production-code replays verify the practical resampling probabilities.

Table 4: **Local approximation error across domains.** Each row summarizes 24 partially generated sequences with three remaining MASK positions.

Domain	Gate	Samples	Median TV	Mean TV
DNA	Soft	24	0.00041	0.00286
DNA	Hard	24	0.00012	0.00968
Protein	Soft	24	0.00073	0.09231
Protein	Hard	24	0.00031	0.03408
RNA	Soft	24	0.02280	0.03571
RNA	Hard	24	0.00123	0.01503

Tab. 4 shows median TV below 0.001 for DNA and protein under both gate modes and below 0.023 for RNA, indicating close agreement for typical sampled updates in this near-terminal regime.

D Why the Hard-Gate Penalty Is Finite

The log-space aggregation in Eq. (10) gives the terminal weight

$$\exp\big(R(x)/\beta\big) = \prod_{k=1}^{n} \Big[\exp\big(g_k(d_k(x))\big)\Big]^{w_k/\beta}. \tag{33}$$

A soft gate contributes $\sigma(d_k(x)/\tau_k)^{w_k/\beta}$, while a hard gate contributes 1 when satisfied and $e^{-Pw_k/\beta}$ otherwise. Thus a finite hard penalty discourages violations without excluding them from the support. Two properties of this construction shape the sampler.

A Constant Penalty Cancels. When all candidate completions violate the same hard-gated requirement, its common contribution $-Pw_k/\beta$ cancels during local normalization. Relative weights then depend on the prior and the remaining requirements. A hard gate distinguishes candidates on opposite sides of its threshold, but provides no ranking on either side. A soft gate additionally distinguishes their margins.

The Penalty Is Finite. A finite penalty keeps local resampling well-defined when all candidate completions violate at least one hard-gated requirement, allowing the remaining reward feedback to guide selection. In the limit $P \to \infty$, these candidates receive zero weight and the local weights cannot be normalized. The strength of violation suppression depends on Pw_k/β relative to the other contributions to the log weight, rather than on P alone.

Table 5: **Sampler configuration of the reported DIMOS runs**, one row per task. T is the number of executed denoising steps, M the candidate width, K the branch width, s the split step, β the tilt temperature of Eq. (8), and η the proposal temperature.

Domain	Task	T	M	K	s	β	η
DNA	ATAC	128	4	4	64	0.25	3.0
	MaxGap	128	4	8	64	0.25	3.0
Protein	SeqRec	49	4	2	0	0.02	1.0
	GRAVY	49	2	2	0	0.02	1.0
RNA	Accessibility	49	4	2	18	0.04	0.05
	MFE	49	4	2	18	0.04	1.0

E Algorithm and Implementation Details

In this section, we detail the implementation of DIMOS. We first specify the sampling schedule and transition notation, then present the full algorithm and explain how completions are used for local guidance and branch scoring.

E.1 Sampling Schedule and Transition Notation

Algorithm 1 describes local enumeration and branch search for the main-text experiments. We index denoising steps by $i = 0, \ldots, L-1$, so branching occurs before the first update when $s = 0$. The corresponding noise index is $\ell = L - i$. Token enumeration is enabled from the first guided update, independently of the split step. The configured guidance window is $\mathcal{G}$; branch-selection checkpoints are $\mathcal{C} = \{i : s < i < L-1,\ (i-s) \bmod f = 0\}$, where f is the search interval. The final update is unguided in the default configurations.

Reveal returns a shared base successor z and its newly revealed positions E: it chooses a count using the configured reveal schedule, selects that many remaining masked positions uniformly, and samples their tokens. Previously committed tokens stay fixed. We write q_i^j for the actual conditional token distribution after proposal tempering, and Q_i for the corresponding IID transition. For enumeration, all candidates share z except at the selected position, so only the factor $q_i^j(a_m)$ varies in their prior weights. When E is empty, the code uses IID candidates from Q_i and reward-only weights, without multiplying their proposal probability a second time.

E.2 DIMOS Algorithm

E.3 Completion and Branch Reward Estimation

Complete preserves committed tokens and fills only remaining masks using the configured completion rule: stochastic completion with shared random numbers across a parent's candidates for DNA, and argmax completion for protein/RNA. Denoiser outputs are reused for branch reward estimation where available; a reward evaluation is reused only when the completion is the same. Branch selection uses an interval of $f = 1$.

F Evaluation Criteria

In this section, we define the task-specific success thresholds and the metrics used to assess joint feasibility, design quality, diversity, validity, and efficiency.

F.1 Task Success Thresholds

Tab. 6 lists the final evaluation criteria, shared across methods and gate modes within each task.

Algorithm 1 DiMOS with local token enumeration and branch search

Require: frozen denoiser μ_θ; joint reward R; B lineages, L steps, M tokens, K branches; split s; guidance steps $\mathcal{G}$, search checkpoints $\mathcal{C}$; temperatures η, β
Ensure: B clean designs

initialize one fully masked state in each of B independent lineages
for $i = 0, \dots, L-1$ **do**
  $\ell \leftarrow L - i$
  if $i = s$ **and** $K > 1$ **then**
    clone each lineage's state into K branches
  end if
  for all active states x **in parallel do**
    if $i \in \mathcal{G}$ **and** x contains masks **then**
      compute tempered token conditionals q_i from $\mu_\theta(x, t_\ell)$ using η
      $(z, E) \leftarrow \textsc{Reveal}(x, q_i, i)$
      if $E \neq \varnothing$ **then** ▷ enumerated menu
        $j \leftarrow \arg\max_{u \in E} H(q_i^u)$
        $(a_1, \dots, a_M) \leftarrow \mathrm{TopM}(q_i^j)$ ▷ distinct tokens
        $y^{(m)} \leftarrow z^{j \leftarrow a_m}$, $b_m \leftarrow \log q_i^j(a_m)$ for each m
      else ▷ empty-reveal fallback
        draw $y^{(1)}, \dots, y^{(M)}$ independently from $Q_i(\cdot \mid x)$; $b_m \leftarrow 0$
      end if
      $\hat{x}_0^{(m)} \leftarrow \textsc{Complete}(y^{(m)}, t_{\ell-1})$; $R^{(m)} \leftarrow R(\hat{x}_0^{(m)})$
      $\log \widehat{w}_m \leftarrow b_m + R^{(m)}/\beta$
      $m^\star \sim \mathrm{Categorical}(\mathrm{softmax}(\log \widehat{w}_{1:M}))$; $x \leftarrow y^{(m^\star)}$
    else
      advance x with the unguided reverse update
    end if
  end for
  if $K > 1$ **and** $i \in \mathcal{C}$ **then**
    compute each branch's reward as $R(\textsc{Complete}_{\mathrm{argmax}}(x))$
    within each lineage, copy the highest-reward branch into all K slots
  end if
end for
apply the configured final denoising to any residual masks
return the highest-reward clean branch from each lineage

Table 6: **Canonical joint-success criteria for the six design tasks.** Both objective conditions must be satisfied.

Task	Objective 1	Objective 2
Activity + ATAC	Activity ≥ 4.0	ATAC ≥ 0.7
Activity + MaxGap	Activity ≥ 4.0	MaxGap ≥ 1.0
ddG + SeqRec	ddG ≥ 0	SeqRec ≥ 0.55
ddG + GRAVY	ddG ≥ 0	GRAVY ≤ -0.70
MRL + Opening dG	Opening dG ≤ 0.25 kcal/mol	Optimus-MRL ≥ 0.45
MRL + MFE	MFE ≤ -16 kcal/mol	Optimus-MRL ≥ 0.45

F.2 Metric Definitions

❶ **Joint Success Rate (SR)** is the primary metric: the fraction of returned designs satisfying both canonical objective thresholds, rather than optimizing one objective at the expense of the other. For RNA, joint SR uses the held-out Optimus-MRL reward model (Sample et al., 2019) together with the task-specific structural criterion, while generation is guided by ConvGRU-MRL.

❷ **Objective Quality** reports each objective's median over all outputs and its median among the ten designs with the highest joint reward, capturing both distribution-wide steering and best-design quality. Top-10 designs are ranked by the joint reward rather than either objective alone.

❸ **Diversity and Validity** include exact-sequence uniqueness, DNA naturalness, and protein fold recovery. Uniqueness is the percentage of distinct exact sequences. DNA validity uses 3-mer and

JASPAR 2022 motif-profile correlations (Castro-Mondragon et al., 2022) to the top-0.1% HepG2 reference set. Protein validity uses held-out ESMFold (Lin et al., 2023), reporting the fractions with scRMSD < 2Å and pLDDT > 80. These checks expose particle collapse and reward hacking beyond the guidance reward models.

❹ **Efficiency** is generation time per returned design, reported in milliseconds and not ranked. Within any single table all timings come from one measurement batch run with a single process per GPU and with kernel compilation disabled, so that wall clock is not distorted by contention and so that a speed setting never changes a reported result. For BEST-OF-N, we select the tested candidate count whose measured generation time is closest to DIMOS's within each task and gate mode.

All main-table entries are mean $\pm$ population standard deviation over three independent seeds.

G BASELINE AND BUDGET PROTOCOL

We use the pretrained DNA and structure-conditioned protein generators released with DRAKES (Wang et al., 2025). DNA designs are 200-bp sequences in the GOSAI setting (Gosai et al., 2024). Protein generation is conditioned on 12 held-out backbones.

PRETRAINED samples directly from the frozen prior. BEST-OF-N draws independent samples and selects by terminal joint reward; its candidate count is calibrated to the closest DIMOS wall-clock budget. TDS combines a gradient-guided proposal with sequential importance correction. SMC uses the pretrained reverse kernel as its proposal and resamples particles; FK-STEERING uses the published discrete Feynman–Kac difference potential; and SVDD-PM greedily selects local candidates using posterior-mean values. RNA excludes TDS because neither selected task supplies a differentiable full joint reward.

For the gated-reward comparisons, DIMOS and all reward-based baselines share the same reward parameters within each task and gate mode: $P = 30$, $w_k = 1$ for every requirement, and the soft-gate widths in Tab. 7. In soft-gate mode, both requirements use soft gates. In hard-gate mode, Activity (DNA), $\Delta\Delta G$ (protein), or MRL (RNA) uses the binary $0/-P$ gate, while the other requirement retains its soft gate. The width τ_k is unused for a hard gate. PRETRAINED does not use rewards during generation.

Table 7: **Shared soft-gate widths.** The widths follow the reward order shown.

Domain	Reward Pair	τ_1	τ_2
DNA	Activity + ATAC	0.5	0.05
	Activity + MaxGap	0.5	0.5
Protein	$\Delta\Delta G$ + SeqRec	0.3	0.03
	$\Delta\Delta G$ + GRAVY	0.3	0.10
RNA	MRL + Opening ΔG	0.05	0.25
	MRL + MFE	0.05	1.0

Every baseline runs at the defaults published with its own implementation. BEST-OF-N and SVDD expose a compute budget through their candidate counts K and M, which we set per block to the closest match to the DIMOS generation budget that is not cheaper than it.

Every method returns the same number of final designs within each domain: $N = 640$ DNA sequences per seed, $N = 128$ protein sequences per wild-type backbone, and $N = 128$ RNA sequences per seed. Task definitions, frozen priors, and reward models used for evaluation are fixed across methods. DNA activity and protein ddG rewards are computed using model checkpoints separate from those used for guidance. Recorded generation wall time is normalized by the number of returned designs.

H SEPARATING THE SEARCH LAYER FROM THE LOCAL KERNEL

Tab. 3 removes components from DIMOS. This section runs the complementary test: it gives the search layer of Sec. 4.3 to a different local kernel. We keep SVDD-PM at its released hyperparameters

Table 8: **Separating the search layer from the local kernel.** DNA activity–ATAC, $N = 640$, three seeds. Rows marked Tab. 1 are reproduced from the main results. Oracle counts reward-model forward passes per seed; time is generation wall clock per returned design.

Method	M	K	s	Joint SR (%) ↑	3-mer r	Oracle	Time (ms)
Soft-Gate Mode							
SVDD-PM (Tab. 1)	12	1	–	55.16±1.55	0.74	1.97M	244.7
+ search, matched budget	4	4	64	63.02±2.20	0.73	1.96M	251.0
+ search, released M	20	4	64	84.43±0.96	0.79	8.51M	1101.8
DIMOS (Tab. 1)	4	4	64	98.39±0.29	0.88	1.94M	234.6
Hard-Gate Mode							
SVDD-PM (Tab. 1)	12	1	–	45.05±0.97	0.69	1.97M	244.6
+ search, matched budget	4	4	64	52.34±0.80	0.65	1.96M	250.7
+ search, released M	20	4	64	72.86±1.41	0.71	8.51M	1101.8
DIMOS (Tab. 1)	4	4	64	87.55±0.65	0.86	1.94M	234.6

Table 9: **SVDD-PM candidate width on DNA activity–ATAC.** $T = 128$, $N = 640$, three seeds, no search layer. Oracle and time are shared by the two gate modes to within 0.2%.

M	Oracle	Time (ms)	Joint SR (%) ↑		3-mer r	
			Soft	Hard	Soft	Hard
4	0.66M	84	26.30±0.83	18.59±1.38	0.69	0.64
8	1.31M	159	46.25±1.00	37.71±1.36	0.72	0.68
10	1.64M	184	51.51±1.67	43.07±0.70	0.73	0.69
12 (Tab. 1)	1.97M	245	55.16±1.55	45.05±0.97	0.74	0.69
16	2.62M	336	60.21±2.78	51.56±1.67	0.75	0.71
20	3.28M	434	63.70±1.09	55.16±1.85	0.76	0.71

and change only where computation is spent. Branches are scored with SVDD-PM's own posterior-mean completion, so no component of Sec. 4.2 enters. Solving the search arm's cost for the DIMOS oracle budget gives $M = 4$, which is also DIMOS's own candidate width, so the matched row and DIMOS differ only in the per-step transition.

The search layer transfers. At $1.03\times$ the SVDD-PM generation time and a slightly smaller oracle budget, it adds 7.86 and 7.29 points on the soft and hard gates.

The local kernel accounts for the larger share. The matched row and DIMOS share M, K, and s and differ only in how one step selects its next state, and that difference is 35.37 and 35.21 points. Raising the SVDD candidate width until the search arm spends $4.39\times$ the DIMOS oracle budget and $4.70\times$ its time still leaves it 13.96 and 14.69 points behind. Naturalness moves the same way: the transferred search costs the SVDD kernel some 3-mer correlation (0.74 to 0.73 soft, 0.69 to 0.65 hard), while DIMOS attains the highest value in both modes.

Tab. 9 places these arms against the only budget knob the SVDD kernel has on its own. Joint success rises monotonically with the candidate width M, but the marginal return falls from 5.0 points per unit M between $M = 4$ and $M = 8$ to 0.9 between $M = 16$ and $M = 20$, so buying joint feasibility through candidates alone becomes expensive well before the gap to DIMOS closes. The matched search arm comes within 0.7 points of the $M = 20$ row on the soft gate while spending $0.60\times$ its oracle budget and $0.58\times$ its time, and lands near $M = 17$ on the hard gate. Reallocating a fixed budget is therefore worth more to this kernel than widening its candidate set.

I Additional Tasks

In this section, we extend the evaluation to DNA cell-type specificity, protein hydrophilicity, and RNA translation paired with accessibility or folding stability. We examine these additional requirements under both gate modes.

Table 10: **DNA cell-type specificity: *Activity* + *MaxGap*.** HepG2 activity paired with a specificity margin against K562 and SK-N-SH. Unlike *Activity* + *ATAC* the two objectives are in tension. The activity term uses a hard gate or a soft gate, while MaxGap retains its soft gate. With a hard gate, activity's own pass rate *falls* and the designs drift toward uniform composition. Uniqueness, 3-mer r and JASPAR r are guardrails, not objectives, and are reported without ranking. $N = 640$ per seed, mean $\pm$ s.d. over three seeds; time is ms per design.

Method	Joint SR (%) ↑	Activity ↑		MaxGap ↑		Uniq. (%)	3-mer r	JASPAR r	Time (ms)
		Median	Top-10	Median	Top-10				
Hard-Gate Mode									
Pretrained	0.00±0.00	0.18±0.01	2.84±0.25	-0.12±0.00	0.39±0.30	100.00±0.00	-0.04±0.01	0.29±0.01	8.48±0.20
Best-of-N	1.20±0.27	1.16±0.05	4.49±0.24	0.44±0.00	2.37±0.65	100.00±0.00	0.40±0.01	0.74±0.00	273.38±0.32
TDS	7.71±4.23	**4.37±0.10**	5.23±0.20	-0.37±0.20	1.88±0.45	46.30±1.62	0.77±0.06	0.85±0.04	171.94±0.70
SMC	5.89±1.42	4.35±0.10	4.79±0.13	-0.44±0.07	2.11±0.49	47.34±3.77	0.79±0.03	0.89±0.01	75.43±0.19
FK-Steering	1.72±0.34	0.59±0.02	4.61±0.11	0.19±0.01	3.02±0.47	100.00±0.00	0.18±0.04	0.52±0.03	69.84±0.61
SVDD	35.16±1.60	3.71±0.06	6.55±0.09	1.62±0.07	5.59±0.35	100.00±0.00	0.46±0.01	0.74±0.01	280.22±0.44
DiMOS (Ours)	**57.03±0.92**	4.34±0.02	**7.54±0.07**	**3.11±0.12**	**6.68±0.29**	100.00±0.00	0.26±0.01	0.74±0.01	275.35±0.07
Soft-Gate Mode									
Pretrained	0.00±0.00	0.18±0.01	2.84±0.25	-0.12±0.00	0.39±0.30	100.00±0.00	-0.04±0.01	0.29±0.01	8.48±0.20
Best-of-N	1.25±0.22	2.40±0.03	4.49±0.23	0.07±0.04	2.37±0.65	100.00±0.00	0.49±0.03	0.78±0.02	274.49±1.16
TDS	9.79±1.96	2.95±0.03	5.40±0.67	0.21±0.24	3.32±0.33	46.98±0.70	0.28±0.10	0.42±0.12	167.61±1.22
SMC	11.09±5.18	2.71±0.55	5.34±0.29	0.21±0.12	3.08±0.41	48.80±0.48	0.34±0.24	0.53±0.19	75.09±0.22
FK-Steering	1.72±0.46	1.63±0.11	4.62±0.14	-0.01±0.02	2.96±0.49	100.00±0.00	0.39±0.04	0.71±0.03	69.80±0.26
SVDD	44.06±2.13	4.05±0.04	7.10±0.13	1.56±0.07	4.83±0.39	100.00±0.00	0.60±0.00	0.84±0.00	280.49±0.39
DiMOS (Ours)	**72.60±0.29**	**5.06±0.05**	**7.69±0.11**	**2.26±0.10**	**5.50±0.25**	100.00±0.00	0.73±0.01	0.87±0.00	275.73±0.42

I.1 DNA Cell-Type Specificity

Tab. 10 reports the second DNA task, *Activity + MaxGap*, in which HepG2 activity is paired with a specificity margin against K562 and SK-N-SH. Unlike *Activity + ATAC*, the two objectives are in tension: in the reference enhancer set activity and specificity are negatively correlated, so the direction that raises expression in HepG2 tends to raise it in the off-target lines as well. The hard-gate mode replaces activity's smooth margin feedback with a binary penalty, removing information about how far a candidate lies from the activity threshold.

I.2 Protein Hydrophilicity (GRAVY)

Tab. 11 repeats the protein comparison with the secondary objective replaced by hydrophilicity, GRAVY < -0.70, using the Kyte–Doolittle hydropathy scale (Kyte & Doolittle, 1982) under the same two gate modes. Two features of this task are worth isolating.

First, TDS reaches 0.00% joint success in both gate modes, despite clearing the binding threshold on approximately 52–54% of its designs. No output satisfies the hydrophilicity requirement.

Second, the wall-clock axis runs the other way here. DiMOS costs about 80 ms per design in both gate modes, *below* every published baseline budget, so the budget-matched partners are Best-of-N at $K{=}8$ and SVDD at $M{=}4$ rather than the larger settings used elsewhere. At that budget DiMOS reaches 27.39% against 18.73% and 19.12%, and it also exceeds the strongest baseline at *any* budget we measured. Both GRAVY configurations use $M{=}2$ and $K{=}2$.

I.3 RNA Design

We additionally evaluate RNA design (Lorenz et al., 2011; Li et al., 2025) on translation paired with accessibility or folding stability. The two gate modes differ in the translation term: it uses a hard gate or a soft gate, while the secondary objective retains its soft gate. Both modes use the same success criteria.

Translation and Accessibility. Tab. 12 is the RNA translation + accessibility result, moved here for space. Both objectives use separate reward models for guidance and final evaluation: guidance uses a ConvGRU MRL reward model and a constrained-MFE opening proxy, while the table reports Optimus-5′ MRL and the exact partition-function opening energy. The gap between the two reveals

Table 11: **Protein binder design, $\Delta\Delta G > 0$ and GRAVY < -0.70.** Same gate modes and protocol as Table 2. `Best-of-N` and `SVDD` are shown at the budget closest to DIMOS's wall clock, which in the hard-gate block is *below* their published budgets because DIMOS runs at 79 ms per design there. Time is milliseconds per design.

Method	Joint SR (%) ↑	ΔΔG ↑		GRAVY ↓		Uniq. (%)	scRMSD (%)	pLDDT (%)	Time (ms)
		Median	Top-10	Median	Top-10				
Hard-Gate Mode									
`Pretrained`	6.84±0.21	-0.45±0.00	0.28±0.02	-0.57±0.00	-0.77±0.00	99.67±0.05	93.55±0.11	87.98±0.11	16.90±0.08
`Best-of-N`	18.73±0.16	-0.28±0.00	0.44±0.02	-0.75±0.00	-0.86±0.00	98.31±0.09	93.55±0.05	87.96±0.14	81.64±0.09
`TDS`	0.00±0.00	**0.18**±0.03	0.36±0.04	0.43±0.01	0.24±0.01	11.11±0.27	88.76±0.35	84.38±0.00	121.42±0.75
`SMC`	16.93±1.24	-0.28±0.01	0.40±0.06	-0.51±0.00	-0.75±0.01	22.59±0.77	94.92±0.63	86.65±0.37	80.37±0.06
`FK-Steering`	12.89±0.14	-0.28±0.00	**0.51**±0.01	-0.64±0.00	-0.79±0.01	99.18±0.11	93.06±0.16	87.48±0.11	192.31±0.03
`SVDD`	19.12±0.27	-0.61±0.03	0.33±0.01	-0.95±0.00	-1.00±0.01	99.09±0.09	92.58±0.09	88.00±0.27	85.79±0.02
DIMOS (Ours)	**27.39**±0.12	-0.71±0.02	0.40±0.02	**-1.08**±0.00	**-1.05**±0.00	100.00±0.00	91.86±0.23	84.85±0.13	79.98±0.19
Soft-Gate Mode									
`Pretrained`	6.84±0.21	-0.45±0.00	0.28±0.02	-0.57±0.00	-0.77±0.00	99.67±0.05	93.55±0.11	87.98±0.11	16.73±0.10
`Best-of-N`	21.83±0.27	-0.28±0.00	0.49±0.01	-0.76±0.00	-0.86±0.00	98.63±0.09	91.60±0.11	88.19±0.16	81.48±0.08
`TDS`	0.00±0.00	**0.12**±0.01	0.39±0.05	0.42±0.01	0.22±0.00	10.37±0.21	89.97±0.83	85.42±0.00	117.91±0.01
`SMC`	10.13±0.35	-0.43±0.02	0.27±0.04	-0.60±0.01	-0.81±0.00	23.61±0.91	93.55±0.23	87.48±0.52	78.98±0.17
`FK-Steering`	13.41±0.09	-0.41±0.00	0.35±0.02	-0.71±0.01	-0.86±0.00	99.26±0.03	92.25±0.11	88.06±0.08	192.13±0.08
`SVDD`	36.22±0.26	-0.15±0.02	0.54±0.01	-0.86±0.00	-0.96±0.00	97.55±0.08	90.47±0.03	88.89±0.03	85.53±0.03
DIMOS (Ours)	**46.94**±0.05	-0.03±0.01	**0.84**±0.03	**-0.98**±0.00	**-1.02**±0.00	99.74±0.05	91.04±0.13	85.50±0.06	80.35±0.47

Table 12: **RNA 5′UTR design with hard and soft gates.** Both blocks are the same *translation* + *accessibility* task with the same success criterion, and differ only in whether the translation term uses a hard gate (binary penalty) or a soft gate (smooth reward). $N=128$ designs per seed. **Both objectives are graded by held-out oracles**: guidance sees a ConvGRU MRL predictor and a constrained-MFE opening proxy, while the table reports Optimus-5′ MRL and the exact partition-function opening energy $-RT \log p_{\text{unpaired}}$, a different physical quantity rather than a finer estimate of the proxy. Each baseline is shown at its measured time-closest budget rung; none runs below our budget. Mean ± s.d. over three seeds; time is milliseconds per design.

Method	Joint SR (%) ↑	Opening ΔG ↓		ConvGRU-MRL ↑		Uniq. (%)	Optimus-MRL ↑	Time (ms)
		Median	Top-10	Median	Top-10			
Hard-Gate Mode								
`Pretrained`	0.52±0.45	4.07±0.17	2.01±0.40	0.34±0.08	0.81±0.03	100.00±0.00	-0.11±0.11	10.69±0.08
`Best-of-N`	1.04±1.19	2.04±0.08	0.73±0.03	0.89±0.01	0.88±0.00	100.00±0.00	0.55±0.02	186.22±0.32
`FK-Steering`	1.82±1.80	2.25±0.16	0.77±0.26	0.81±0.01	0.89±0.02	100.00±0.00	0.36±0.04	725.52±9.83
`SMC`	2.86±4.30	0.93±0.13	0.51±0.10	0.93±0.06	1.01±0.10	28.39±1.19	0.62±0.04	202.26±8.12
`SVDD`	1.82±0.90	1.01±0.09	0.44±0.02	0.86±0.01	0.97±0.02	100.00±0.00	0.50±0.02	347.65±3.13
DIMOS (Ours)	**84.11**±3.69	**0.13**±0.00	**0.06**±0.01	**0.98**±0.01	**1.08**±0.01	100.00±0.00	**0.74**±0.01	178.24±1.39
Soft-Gate Mode								
`Pretrained`	0.52±0.45	4.07±0.17	2.01±0.40	0.34±0.08	0.81±0.03	100.00±0.00	-0.11±0.11	11.25±0.88
`Best-of-N`	1.30±1.63	1.43±0.08	0.65±0.02	0.82±0.01	0.88±0.01	100.00±0.00	0.46±0.00	189.46±1.19
`FK-Steering`	0.26±0.45	2.34±0.18	0.73±0.04	0.77±0.02	0.92±0.02	100.00±0.00	0.35±0.01	726.31±7.12
`SMC`	0.00±0.00	0.66±0.25	0.52±0.04	0.88±0.07	0.96±0.01	34.11±3.16	0.40±0.16	202.14±4.76
`SVDD`	4.17±1.19	0.91±0.13	0.34±0.05	**1.07**±0.01	**1.21**±0.01	100.00±0.00	0.69±0.02	293.86±4.65
DIMOS (Ours)	**78.39**±4.96	**0.14**±0.01	**0.08**±0.01	**1.07**±0.00	1.13±0.01	100.00±0.00	**0.81**±0.01	172.14±1.06

discrepancies between proxy and exact structural rewards. In hard-gate mode, reward-guided baselines pass the opening proxy criterion on 14–77% of designs but the exact criterion on only 1.6–4.7%, while the corresponding rates for DIMOS are 99.7% and 85.2%.

For DIMOS, replacing the translation soft gate with a hard gate raises joint SR from 78.39% to 84.11%. The held-out translation pass rate remains high (99.74% versus 98.96%), while the exact accessibility pass rate rises from 78.65% to 85.16%. This is consistent with directing more reward feedback toward the remaining structural requirement.

Translation and Folding Stability. Tab. 13 repeats the RNA comparison with accessibility replaced by the 50-nt minimum free energy. Only the translation objective uses a held-out reward model: MRL uses a ConvGRU reward model for guidance and Optimus-5′ for evaluation, while the folding energy has no cheaper proxy and is therefore computed by the same ViennaRNA call on both sides.

Table 13: **RNA 5′UTR design, translation + folding stability.** As in Table 12, with accessibility replaced by the 50-nt minimum free energy. MRL is graded by the held-out Optimus-5′ predictor; MFE has no cheaper guidance proxy and is the one objective shared between guidance and grading. Mean ± s.d. over three seeds; time is milliseconds per design.

Method	Joint SR (%) ↑	MFE ↓		ConvGRU-MRL ↑		Uniq. (%)	Optimus-MRL ↑	Time (ms)
		Median	Top-10	Median	Top-10			
Hard-Gate Mode								
Pretrained	0.26±0.45	-8.57±0.53	-12.13±1.69	0.34±0.08	0.75±0.02	100.00±0.00	-0.11±0.11	10.73±0.23
Best-of-N	0.78±1.35	-9.38±0.23	-12.72±0.88	**0.85±0.01**	0.84±0.01	100.00±0.00	**0.47±0.02**	78.81±2.70
FK-Steering	2.08±0.45	-11.39±0.18	-14.29±1.16	0.72±0.03	0.84±0.01	100.00±0.00	0.32±0.06	140.91±1.77
SMC	18.75±7.53	-14.77±0.89	-16.10±0.62	**0.85±0.01**	0.85±0.02	21.09±4.13	0.46±0.02	68.72±0.93
SVDD	15.62±2.07	-14.09±0.21	-18.40±0.74	0.83±0.00	0.86±0.01	100.00±0.00	0.44±0.03	74.88±3.49
DiMOS (Ours)	**36.72±2.07**	**-18.36±0.17**	**-20.33±0.27**	0.82±0.01	**0.88±0.03**	100.00±0.00	0.46±0.02	36.12±4.43
Soft-Gate Mode								
Pretrained	0.26±0.45	-8.57±0.53	-12.13±1.69	0.34±0.08	0.75±0.02	100.00±0.00	-0.11±0.11	10.92±0.43
Best-of-N	1.30±0.90	-11.32±0.43	-14.43±0.36	0.77±0.02	0.77±0.01	100.00±0.00	0.41±0.02	79.33±1.58
FK-Steering	4.43±2.26	-11.62±0.33	-15.35±0.53	0.74±0.03	0.83±0.02	100.00±0.00	0.36±0.03	140.44±2.46
SMC	21.35±19.09	-17.19±0.60	-17.47±0.90	0.77±0.05	0.87±0.02	26.04±2.39	0.34±0.14	70.06±3.43
SVDD	19.27±1.63	-14.21±0.64	-18.31±0.45	0.88±0.01	0.91±0.02	100.00±0.00	**0.51±0.03**	68.83±1.27
DiMOS (Ours)	**54.17±0.45**	**-18.72±0.14**	**-20.10±0.42**	**0.92±0.00**	**0.99±0.01**	100.00±0.00	0.50±0.02	35.71±3.53

This task benefits from graded translation feedback: DiMOS's soft-gate guidance achieves 54.17% joint SR and 62.50% held-out translation success, compared with 36.72% and 51.30% under hard-gate guidance. Together with the accessibility results, this shows that the effect of the gate depends on which requirements remain difficult to satisfy.

J Continuous-Score Rewards

Sec. 4.1 builds the joint reward from per-property thresholds. The local resampling rule of Sec. 4.2 uses $\exp(R/\beta)$ and applies directly to continuous property scores that continue to increase beyond the acceptance thresholds. Tabs. 14 and 15 repeat the DNA and protein tasks with the gated reward of Eq. (10) replaced by a weighted sum of the two property scores, standardized on frozen-prior samples. The task, the seven configurations and the success criteria are unchanged, and nothing was re-tuned.

Table 14: **DNA design guided by a continuous property score.** The same *Activity + ATAC* task and the same seven configurations as Tab. 1, with the gated reward of Eq. (10) replaced by a weighted sum of the two property scores standardized on frozen-prior samples, $R_\lambda = \lambda z_{\text{act}} + (1-\lambda) z_{\text{ATAC}}$. No acceptance threshold enters the reward, and nothing was re-tuned. Joint SR is scored against the unchanged canonical criteria. λ weights the first objective and is part of the reward, not a sampler setting: at one λ every method optimizes the same R_λ. Because z is in units of the prior's own standard deviation, $\lambda = 0.5$ means equal weight in prior spread, not equal importance. Blocks are different objective functions and are not comparable to each other. Guardrails are reported without ranking. $N = 640$ per seed, mean $\pm$ s.d. over three seeds.

Method	Joint SR (%) ↑	Activity ↑		ATAC ↑		Uniq. (%)	3-mer r	JASPAR r	Time (ms)
		Median	Top-10	Median	Top-10				
$\lambda = 0.25$: ATAC-leaning									
Pretrained	0.00±0.00	0.18±0.01	2.41±0.39	0.03±0.00	0.65±0.06	100.00±0.00	-0.04±0.01	0.29±0.01	8.13±0.01
Best-of-N	1.77±0.39	1.51±0.03	5.24±0.26	0.41±0.01	0.89±0.03	100.00±0.00	0.78±0.00	0.91±0.00	202.12±0.10
TDS	3.70±1.67	1.11±0.11	5.28±0.13	0.28±0.05	0.84±0.06	51.88±1.38	0.37±0.06	0.56±0.09	204.19±0.26
SMC	6.25±4.69	1.35±0.29	4.79±0.79	0.29±0.17	0.92±0.01	51.98±1.19	0.39±0.09	0.67±0.08	90.12±0.31
FK-Steering	6.25±0.44	1.47±0.04	5.83±0.22	0.38±0.02	0.95±0.01	100.00±0.00	0.71±0.01	0.89±0.00	110.83±0.59
SVDD	47.34±2.67	4.12±0.13	7.23±0.13	0.90±0.00	**0.99±0.00**	100.00±0.00	0.68±0.01	0.91±0.01	203.62±0.50
DiMOS (Ours)	**89.90±1.19**	**5.75±0.05**	**7.52±0.18**	**0.99±0.00**	**0.99±0.00**	100.00±0.00	0.91±0.01	0.93±0.00	195.46±0.74
$\lambda = 0.50$: equal weight									
Pretrained	0.00±0.00	0.18±0.01	2.41±0.39	0.03±0.00	0.65±0.06	100.00±0.00	-0.04±0.01	0.29±0.01	8.34±0.10
Best-of-N	1.77±0.39	2.06±0.01	5.24±0.26	0.37±0.01	0.89±0.03	100.00±0.00	0.77±0.01	0.90±0.00	207.86±0.22
TDS	6.61±1.98	1.05±0.17	5.82±0.78	0.24±0.09	0.89±0.04	53.12±2.01	0.57±0.19	0.66±0.13	209.51±0.40
SMC	7.40±3.94	1.29±0.38	5.56±0.37	0.16±0.03	0.85±0.07	52.76±0.70	0.55±0.04	0.73±0.04	89.10±0.24
FK-Steering	7.71±0.53	2.03±0.05	5.88±0.10	0.33±0.01	0.94±0.01	100.00±0.00	0.70±0.02	0.90±0.00	110.64±0.39
SVDD	49.84±1.33	4.63±0.12	7.26±0.04	0.82±0.00	0.98±0.00	100.00±0.00	0.76±0.00	0.93±0.00	203.46±0.42
DiMOS (Ours)	**92.03±0.46**	**5.85±0.03**	**7.40±0.13**	**0.98±0.00**	**0.99±0.00**	100.00±0.00	0.93±0.01	0.94±0.00	195.50±0.76
$\lambda = 0.75$: Activity-leaning									
Pretrained	0.00±0.00	0.18±0.01	2.41±0.39	0.03±0.00	0.65±0.06	100.00±0.00	-0.04±0.01	0.29±0.01	8.28±0.01
Best-of-N	1.77±0.39	2.38±0.02	5.24±0.26	0.25±0.00	0.89±0.03	100.00±0.00	0.68±0.02	0.85±0.01	202.70±0.22
TDS	8.54±4.41	2.03±0.76	5.49±0.14	0.18±0.07	0.90±0.06	53.33±0.70	0.57±0.21	0.65±0.11	208.40±0.17
SMC	5.52±5.22	1.13±0.28	5.42±0.35	0.13±0.02	0.87±0.09	52.50±0.34	0.39±0.14	0.61±0.14	88.47±0.32
FK-Steering	6.61±0.32	2.10±0.09	5.94±0.17	0.20±0.01	0.94±0.02	100.00±0.00	0.62±0.03	0.85±0.02	110.55±0.38
SVDD	38.70±1.47	4.98±0.02	7.17±0.05	0.62±0.03	0.98±0.00	100.00±0.00	0.77±0.00	0.94±0.00	203.41±0.40
DiMOS (Ours)	**91.77±0.52**	**5.89±0.00**	**7.60±0.11**	**0.97±0.00**	**0.99±0.00**	100.00±0.00	0.95±0.00	0.94±0.00	195.48±0.72

Table 15: **Protein design guided by a continuous property score.** The same $\Delta\Delta G$ + *SeqRec* task and the same seven configurations as Tab. 2, with the gated reward replaced by $R_\lambda = \lambda z_{\Delta\Delta G} + (1-\lambda) z_{\text{SeqRec}}$ on frozen-prior statistics. No acceptance threshold enters the reward, and nothing was re-tuned. Joint SR is scored against the unchanged $\Delta\Delta G > 0$ and SeqRec > 0.55. λ weights $\Delta\Delta G$ and is part of the reward, not a sampler setting. Because z is in units of the prior's own standard deviation, $\lambda = 0.5$ means equal weight in prior spread, not equal importance. Blocks are different objective functions and are not comparable to each other. Guardrails are reported without ranking. $N = 128$ per backbone over 12 backbones, mean $\pm$ s.d. over three seeds.

Method	Joint SR (%) ↑	$\Delta\Delta G$ ↑		SeqRec ↑		Uniq. (%)	scRMSD (%)	pLDDT (%)	Time (ms)
		Median	Top-10	Median	Top-10				
$\lambda = 0.25$: *SeqRec-leaning*									
Pretrained	4.04±0.30	-0.45±0.00	0.49±0.03	0.44±0.00	0.48±0.00	99.67±0.05	93.55±0.11	87.98±0.11	17.36±0.04
Best-of-N	15.30±0.19	0.03±0.01	0.80±0.02	0.50±0.00	0.52±0.00	99.15±0.16	93.36±0.16	86.46±0.16	160.54±0.12
TDS	3.84±0.00	-0.47±0.04	0.61±0.11	0.45±0.00	0.47±0.00	21.70±0.72	95.25±0.42	89.34±0.15	126.66±0.03
SMC	6.60±0.43	-0.40±0.02	0.42±0.01	0.45±0.00	0.48±0.00	23.65±0.80	93.47±0.62	86.98±0.14	84.37±0.84
FK-Steering	4.38±0.06	-0.46±0.00	0.57±0.02	0.45±0.00	0.49±0.00	99.72±0.08	93.29±0.19	86.65±0.11	193.09±0.01
SVDD	33.59±0.33	0.32±0.01	0.98±0.01	0.54±0.00	0.57±0.00	97.29±0.22	95.51±0.18	84.24±0.05	167.18±0.10
DiMOS (Ours)	**57.60**±0.27	**0.42**±0.00	**1.16**±0.02	**0.61**±0.00	**0.66**±0.00	100.00±0.00	92.56±0.03	83.92±0.05	158.24±0.02
$\lambda = 0.50$: *equal weight*									
Pretrained	4.04±0.30	-0.45±0.00	0.58±0.02	0.44±0.00	0.47±0.00	99.67±0.05	93.55±0.11	87.98±0.11	17.50±0.02
Best-of-N	15.45±0.11	0.14±0.01	0.92±0.01	0.48±0.00	0.50±0.00	99.35±0.16	93.45±0.06	86.02±0.11	160.49±0.15
TDS	4.36±0.98	-0.46±0.06	0.72±0.06	0.44±0.00	0.46±0.00	22.24±0.20	94.12±0.35	88.00±0.55	125.66±0.16
SMC	5.95±0.34	-0.55±0.06	0.34±0.03	0.44±0.01	0.48±0.00	23.39±0.93	92.62±0.85	86.48±0.46	83.71±0.59
FK-Steering	4.36±0.16	-0.43±0.00	0.63±0.01	0.45±0.00	0.48±0.00	99.39±0.03	92.90±0.16	87.22±0.08	193.28±0.04
SVDD	31.58±0.11	0.46±0.00	1.15±0.00	0.53±0.00	0.55±0.00	97.46±0.05	94.23±0.17	84.57±0.05	166.98±0.10
DiMOS (Ours)	**46.05**±0.41	**0.58**±0.01	**1.28**±0.00	**0.57**±0.00	**0.61**±0.00	99.93±0.05	93.29±0.05	84.74±0.08	162.25±5.94
$\lambda = 0.75$: $\Delta\Delta G$*-leaning*									
Pretrained	4.04±0.30	-0.45±0.00	0.64±0.02	0.44±0.00	0.47±0.00	99.67±0.05	93.55±0.11	87.98±0.11	17.28±0.06
Best-of-N	13.65±0.08	0.24±0.00	1.01±0.00	0.46±0.00	0.49±0.00	99.13±0.13	93.88±0.18	85.83±0.16	159.62±0.13
TDS	5.88±0.30	-0.34±0.01	0.54±0.03	0.44±0.00	0.47±0.00	23.81±0.86	91.51±0.39	88.85±0.06	119.42±0.17
SMC	2.99±0.51	-0.56±0.01	0.44±0.04	0.44±0.00	0.47±0.01	24.76±1.25	93.90±0.13	87.65±0.34	80.04±0.30
FK-Steering	5.92±0.00	-0.36±0.00	0.66±0.01	0.45±0.00	0.47±0.00	99.22±0.11	93.29±0.23	87.54±0.20	192.22±0.01
SVDD	16.91±0.16	0.57±0.01	1.31±0.01	0.49±0.00	0.52±0.00	98.03±0.20	95.59±0.21	85.05±0.17	166.82±0.10
DiMOS (Ours)	**19.51**±0.27	**0.76**±0.01	**1.40**±0.01	**0.50**±0.00	**0.54**±0.00	100.00±0.00	93.90±0.12	85.26±0.03	158.06±0.07