D-DOIT: Training-free Adaptation of Discrete Diffusion via Doob's h-Transform

Jieke Wu[*‡], **Qijie Zhu**[*§], **Weimin Wu**[*†], **Zeqi Ye**[♮], **Minshuo Chen**[♮], **Han Liu**[†§]
[†] Center for Foundation Models and Generative AI, Northwestern University
Department of Computer Science, Northwestern University
[‡] Department of Computer Science, King Abdullah University of Science and Technology
[§] Department of Statistics and Data Science, Northwestern University
[♮] Industrial Engineering & Management Sciences, Northwestern University
jieke.wu@kaust.edu.sa, {qijiezhu2029,wwm,zeqiye2029}@u.northwestern.edu
minshuo.chen@northwestern.edu, hanliu@northwestern.edu

Abstract

We propose D-DOIT (Discrete Doob-Oriented Inference-time Transformation), a training-free and efficient adaptation method for discrete diffusion models with generic rewards. D-DOIT formulates adaptation as sampling from a reward-tilted target distribution and realizes this transport through Doob's h-transform of the discrete diffusion reverse kernel, using only reward values rather than reward gradients. Unlike continuous diffusion, masked discrete diffusion samples categorical token-reveal transitions rather than continuous state updates. D-DOIT derives the corresponding discrete Doob's h-transform, which guides sampling by reweighting reverse transition probabilities instead of adding a drift correction. To make this transformation practical, D-DOIT avoids expensive future rollouts. At each guided step, D-DOIT samples candidate next states, uses the model prediction head to complete each candidate into a clean sequence, evaluates each completion with the reward oracle, and resamples the next state with probabilities proportional to the rewards. An optional late-stage best-of-K refinement further improves sample quality by branching trajectories only near the end of denoising, avoiding the K-fold cost over the full trajectory. Empirically, across regulatory DNA design and protein inverse folding benchmarks, D-DOIT outperforms training-free guidance baselines. It improves enhancer activity and cell-type specificity while preserving sequence naturalness, and achieves the highest success rate in protein inverse folding.

Project Page: https://github.com/WeiminWu2000/D-DOIT.

1 Introduction

We propose D-DOIT (**D**iscrete **D**oob-**O**riented **I**nference-time **T**ransformation), a training-free sampler for reward-guided masked discrete diffusion. Discrete diffusion has become a strong generative modeling paradigm for finite-vocabulary data [1, 2]. Masked discrete diffusion models are especially attractive for biological sequences because they predict clean tokens from partially masked states and provide tractable categorical reverse kernels [3–5]. These models have recently been used for DNA and protein generation [6–8], where the scientific objective is rarely just to sample from the pre-trained distribution. Instead, a practitioner usually wants sequences with high activity, specificity, or other task-level scores for sequence design.

Reward adaptation creates a tension between performance, generality, and cost. Training-based approaches can improve reward optimization, but they update the backbone and often require reward gradients, extra training, and hyperparameter tuning [8, 9]. Training-free methods keep the backbone

[*]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

fixed, but many either assume differentiable rewards or rely on particle and search procedures whose cost grows with trajectory history or search width [10–17]. This motivates the question studied here: can a frozen masked diffusion model be adapted to generic black-box rewards by modifying only its inference-time sampling law?

Our starting point is to view adaptation as a distributional transport problem. Let p_θ be the clean-sequence distribution induced by the frozen masked diffusion reverse chain, and let $R : \mathcal{X} \to \mathbb{R}$ be a sequence-level reward. Rather than optimizing reward in isolation, we target the reward-tilted distribution $\pi_\beta(\mathbf{x}) \propto p_\theta(\mathbf{x}) \exp(R(\mathbf{x})/\beta)$, which keeps the pre-trained model as a sequence prior while favoring high-reward outputs. Doob's h-transform gives the exact local transformation needed to sample from this tilted law. The central object is the downstream value $h(\bar{\mathbf{x}}_{t_l}, t_l)$, the expected terminal reward weight under the frozen reverse chain conditioned on the current partially denoised state.

The discrete setting changes how the transform must be implemented. In continuous diffusion, Doob guidance can be expressed as an additive score or drift correction [18–21]. In masked discrete diffusion, the state space is categorical, generation proceeds through irreversible token-reveal transitions, and there is no Euclidean score field. D-DOIT therefore applies Doob's h-transform at the kernel level: each frozen reverse transition is multiplicatively reweighted by the downstream h-value of the candidate next state. This yields a discrete analogue of the continuous Doob correction while preserving the frozen parameters and allowing black-box reward queries.

The exact transform still requires the future reward-to-go for every candidate reverse state, which is intractable by repeated rollouts. D-DOIT makes the transform executable with two approximations. First, it estimates candidate h-values by using the masked diffusion prediction head to complete each candidate into a clean sequence, then evaluating the reward oracle on that completion. Second, it draws a finite set of candidate reverse states and selects among them with Doob-weighted probabilities. An optional best-of-K refinement branches only after a chosen split update, which hedges irreversible token reveals without multiplying the full denoising trajectory cost. The same sampler gives two operating points: D-DOIT-LBOK-S for a lower-budget regime and D-DOIT-Q for high-budget quality evidence.

We evaluate D-DOIT on regulatory DNA sequence design and protein inverse folding. On the DRAKES HepG2 enhancer benchmark, the near-TDS-speed D-DOIT-LBOK-S matches TDS on predicted activity while substantially improving accessibility, 3-mer correlation, and motif correlation. The high-budget D-DOIT-Q setting reaches 6.19 predicted activity and exceeds training-based DRAKES on Pred-Activity, 3-mer Corr, and JASPAR Corr, while remaining slightly lower on ATAC-Acc. At a similar wall-clock budget, D-DOIT-Q also improves activity, accessibility, and 3-mer correlation over soft value-based decoding (SVDD-PM) and wall-clock-matched endpoint search. Endpoint search achieves higher JASPAR correlation. On the Ctrl-DNA cell-type-specific enhancer benchmark, D-DOIT is a clear training-free method that moves the specificity reward. D-DOIT-Q raises ΔR to 0.119, with a trade-off in motif preservation, and D-DOIT-LBOK-S provides a lower-budget positive-reward point. At a matched wall-clock budget, however, endpoint search captures much of this gain, so the margin is smaller than on HepG2. Guidance with alternative predictors and transfer to held-out evaluators retain the gains over unguided sampling. On protein inverse folding, the same frozen backbone reaches 68.1% joint stability–structure success. These results support D-DOIT as a training-free mechanism for reward-guided sequence generation, not as a matched-cost replacement for every high-budget or training-based method.

In summary, we make three contributions:

- **Discrete Doob h-transform for masked diffusion.** We formulate reward adaptation as sampling from a reward-tilted target distribution and derive the corresponding Doob h-transform of the masked discrete diffusion reverse kernel. The resulting guidance reweights categorical token-reveal transitions, in contrast to the additive drift or score correction used in continuous diffusion. The exact reference transform is kept separate from the executable approximations built on it.
- **Executable finite-candidate sampler.** We make the exact transform practical with a prediction-head h-estimator, finite-candidate Doob-weighted selection, and an optional best-of-K refinement for irreversible reveal decisions. The method uses only reward values and does not update the diffusion backbone. We characterize how tempering, sharpening, and terminal selection deviate from the exact target.

- **Biological validation and trade-offs.** We test the same frozen-backbone sampler on HepG2 enhancer design, cell-type-specific enhancer design, and protein inverse folding. D-DOIT improves over training-free baselines across the reported reward-guided design tasks, while preserving the relevant trade-offs in naturalness, motif structure, and inference budget.

Organization. Section 2 reviews the masked discrete diffusion backbone, defines the reward-tilted target, and states the exact discrete Doob h-transform. Section 3 develops the practical D-DOIT sampler, including the prediction-head h-estimator, finite-candidate weighted update, and optional best-of-K refinement. Section 4 reports experiments on regulatory DNA design, cell-type-specific enhancer design, and protein inverse folding. The appendix provides related work (Section A), the proof of the discrete Doob h-transform (Section B), the standard masked diffusion sampler (Section E), and full experimental details (Sections J, K and M).

Notation. We write $\mathcal{V} = \{A, G, T, C\}$ for the clean token alphabet, $d = |\mathcal{V}|$, and $\mathcal{X} = \mathcal{V}^N$ for length-N DNA sequences. Let m be the mask token, $\widetilde{\mathcal{V}} = \mathcal{V} \cup \{\mathsf{m}\}$, and $\widetilde{\mathcal{X}} = \widetilde{\mathcal{V}}^N$. We use $\mathbf{X}_{t_l}$ for the forward process and $\bar{\mathbf{X}}_{t_l}$ for the learned reverse chain, which runs from the all-MASK state $\bar{\mathbf{x}}_{t_L}$ to a clean sequence $\bar{\mathbf{x}}_0$. We use q for the forward corruption kernel and analytic posterior, P_{θ,t_l} for the learned reverse kernel, p_{θ,t_l} for the reverse-chain marginal, and $R : \mathcal{X} \to \mathbb{R}$ with $\phi(\mathbf{x}_0) = \exp(R(\mathbf{x}_0)/\beta)$ for the black-box reward and reward weight. For the sampler, M denotes the local candidate budget, n_h the number of completions per candidate, τ and γ control proposal and completion tempering, ω sharpens value-based selection, and K denotes the best-of-K branching width.

2 Preliminaries

We first define the frozen masked diffusion reverse chain that serves as the base law for D-DOIT. We then state the reward-tilted target distribution and the discrete Doob h-transform that transports this base law to the target without changing the model parameters [22, 23]. We keep the exact reference transform separate from the executable approximation built on it, which Section 3 develops.

2.1 Masked Discrete Diffusion Models

We specialize to masked discrete diffusion. Let $\mathcal{V}$ be the clean token alphabet, $d = |\mathcal{V}|$, and m a special mask token. Define $\widetilde{\mathcal{V}} := \mathcal{V} \cup \{\mathsf{m}\}$, with sequence spaces $\mathcal{X} = \mathcal{V}^N$ and $\widetilde{\mathcal{X}} = \widetilde{\mathcal{V}}^N$. The data distribution is supported on $\mathcal{X}$, while the forward process evolves on $\widetilde{\mathcal{X}}$ by progressively replacing clean tokens with m. On the grid $t_l = lT/L$, let $b_l \in [0, 1]$ be the masking probability on $[t_{l-1}, t_l]$. The single-token forward kernel is

$$q_{t_{l-1},t_l}(y \mid x) = (1 - b_l)\mathbf{1}\{y = x\} + b_l\mathbf{1}\{y = \mathsf{m}\}, \quad x, y \in \widetilde{\mathcal{V}},$$

so with probability $1 - b_l$ the token stays fixed and otherwise it jumps to the mask state; once masked it stays masked. With $\alpha_l = \prod_{r=1}^{l}(1 - b_r)$, a clean token $x_0 \in \mathcal{V}$ satisfies

$$q_{0,t_l}(y \mid x_0) = \mathbb{P}(X_{t_l} = y \mid X_0 = x_0) = \alpha_l\mathbf{1}\{y = x_0\} + (1 - \alpha_l)\mathbf{1}\{y = \mathsf{m}\}.$$

We choose $\alpha_L \approx 0$ for a sufficiently large T, so that $q_{0,T}$ is concentrated near the mask state. The forward masking process is applied independently across coordinates. The conditional one-step time reversal also has a closed form, and Section F records the forward transition matrices and the analytic reverse kernel that the learned sampler approximates.

Since $\mathbf{x}_0$ is unknown during generation, masked diffusion models approximate the clean-token posterior with a trained neural network. For each coordinate j, let $\mu_\theta^j(\cdot \mid \mathbf{x}_{t_l}, t_l) \in \Delta(\mathcal{V})$ denote the neural network's approximation to $\mathbb{P}(X_0^j = \cdot \mid \mathbf{X}_{t_l} = \mathbf{x}_{t_l})$. Plugging this approximation into the analytic reverse posterior gives the learned one-step reverse transition from t_l to t_{l-1}:

$$P_{\theta,t_l}(x_{t_{l-1}}^j = y \mid \mathbf{x}_{t_l}) = \begin{cases} (1 - \alpha_{l-1})/(1 - \alpha_l), & x_{t_l}^j = \mathsf{m},\ y = \mathsf{m}, \\ (\alpha_{l-1} - \alpha_l)/(1 - \alpha_l)\,\mu_\theta^j(y \mid \mathbf{x}_{t_l}, t_l), & x_{t_l}^j = \mathsf{m},\ y \in \mathcal{V}, \\ \mathbf{1}\{y = x_{t_l}^j\}, & x_{t_l}^j \neq \mathsf{m}. \end{cases} \tag{2.1}$$

Finally, the learned reverse model factorizes over coordinates,

$$P_{\theta,t_l}(\mathbf{x}_{t_{l-1}} \mid \mathbf{x}_{t_l}) = \prod_{j=1}^{N} P_{\theta,t_l}(x^j_{t_{l-1}} \mid \mathbf{x}_{t_l}).$$

Starting from an initial masked state $\bar{\mathbf{X}}_{t_L}$, we iteratively sample $\bar{\mathbf{X}}_{t_{l-1}} \sim P_{\theta,t_l}(\cdot \mid \bar{\mathbf{X}}_{t_l})$ from $l = L$ to $l = 1$ and obtain the final clean sample $\bar{\mathbf{X}}_0$. Let p_{θ,t_l} denote the marginal law of $\bar{\mathbf{X}}_{t_l}$ under this reverse chain, and write $p_\theta = p_{\theta,0}$ for the clean-time law. This learned reverse chain is the frozen base sampler. D-DOIT leaves its parameters fixed and modifies only its reverse transition probabilities at inference time. We next define the target these modified transitions should realize.

2.2 Reward-tilted Target and Doob's h-transform

We now formalize the inference-time adaptation objective. Given a reward function $R : \mathcal{X} \to \mathbb{R}$ and a temperature parameter $\beta > 0$, define the positive reward weight

$$\phi(\mathbf{x}_0) = \exp(R(\mathbf{x}_0)/\beta). \tag{2.2}$$

The desired clean-time target is the reward-tilted version of the frozen model law,

$$\pi_\beta(\mathbf{x}_0) = \frac{p_\theta(\mathbf{x}_0)\phi(\mathbf{x}_0)}{Z_\beta}, \quad Z_\beta = \mathbb{E}_{\mathbf{X}_0 \sim p_\theta}\left[\phi(\mathbf{X}_0)\right]. \tag{2.3}$$

Thus adaptation is a distributional problem: change the terminal law from p_θ to π_β while preserving the frozen reverse chain as the base proposal. The discrete Doob h-function is the object that makes this transport local in time and gives a kernel-level correction.

Definition 2.1 (Discrete Doob h-function)**.** For the frozen reverse chain defined above, the discrete Doob h-function is the downstream reward weight conditioned on a partially denoised state:

$$h(\bar{\mathbf{x}}_{t_l}, t_l) = \mathbb{E}_{p_\theta}\left[\exp(R(\mathbf{X}_0)/\beta) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right], \qquad l = 0, \dots, L. \tag{2.4}$$

At clean time, this reduces to $h(\mathbf{x}_0, 0) = \phi(\mathbf{x}_0)$. At an intermediate time, $h(\bar{\mathbf{x}}_{t_l}, t_l)$ is the reward-to-go under the frozen reverse chain. It therefore measures how promising the current partial sequence is for the terminal reward objective at clean time under the frozen process.

The next theorem states the exact kernel transform induced by this value function. It is the discrete analogue of the continuous Doob correction, but it acts by reweighting transition probabilities rather than by adding a vector field to a continuous score.

Theorem 2.1 (Discrete Doob's h-transform)**.** In the fully masked terminal setting, $\bar{\mathbf{X}}_{t_L}$ is deterministic, so $h(\bar{\mathbf{X}}_{t_L}, t_L) = Z_\beta$ and the terminal distribution is unchanged. Let h be defined as in Definition 2.1. Then Doob's h-transform induces the reverse kernel

$$P^h_{\theta,t_l}(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}) = P_{\theta,t_l}(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}) \frac{h(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1})}{h(\bar{\mathbf{x}}_{t_l}, t_l)}, \qquad l = 1, \dots, L. \tag{2.5}$$

The resulting Doob-transformed reverse chain has marginal distribution

$$p^h_{\theta,t_l}(\bar{\mathbf{x}}_{t_l}) = \frac{p_{\theta,t_l}(\bar{\mathbf{x}}_{t_l})h(\bar{\mathbf{x}}_{t_l}, t_l)}{Z_\beta}, \qquad l = 0, \dots, L,$$

In particular, since

$$h(\mathbf{x}_0, 0) = \exp(R(\mathbf{x}_0)/\beta),$$

the clean-time marginal of the Doob-transformed reverse chain is exactly the reward-tilted distribution

$$p^h_{\theta,0}(\bar{\mathbf{x}}_0) = \pi_\beta(\bar{\mathbf{x}}_0) = \frac{p_{\theta,0}(\bar{\mathbf{x}}_0)\exp(R(\bar{\mathbf{x}}_0)/\beta)}{Z_\beta}.$$

The proof is deferred to Section B. The theorem identifies the exact population object that D-DOIT approximates. If the downstream value ratio $h(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1})/h(\bar{\mathbf{x}}_{t_l}, t_l)$ were available at every reverse step, then local reweighting of the frozen transition kernel would sample exactly from π_β at clean

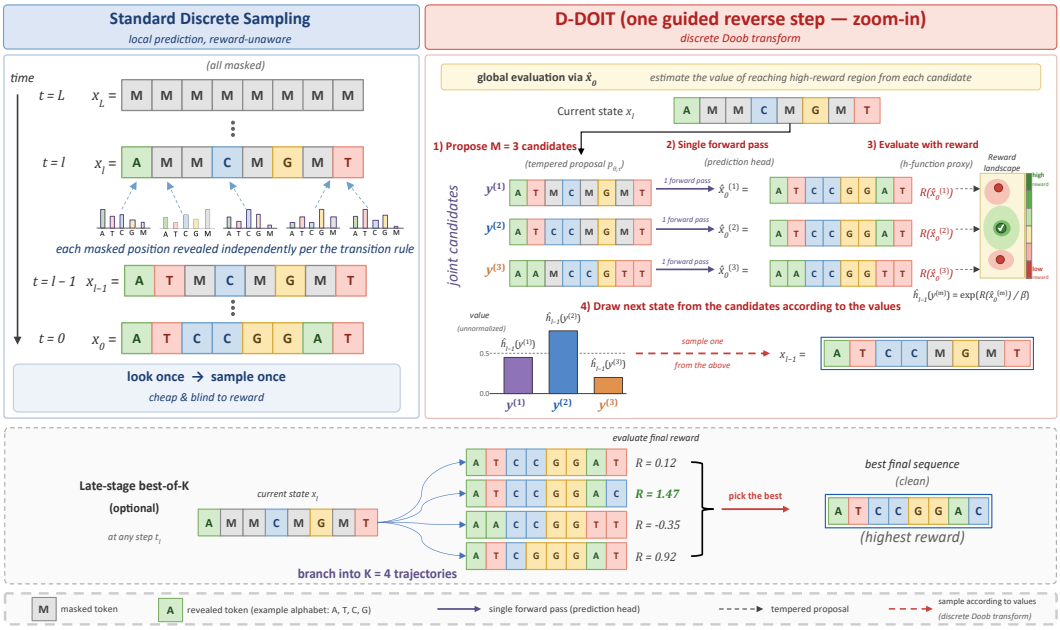


Figure 1: **Overview of D-DOIT. Left:** Standard masked diffusion unmasks positions independently at each reverse step (2.1); it is cheap but reward-unaware. **Right:** One guided step approximately tilts the reverse kernel: it draws $M{=}3$ candidates from a tempered proposal (3.6), completes and scores them with the prediction head and reward to obtain proxy h-values, and selects the next state with sharpened h-weights (3.7). **Bottom:** Optional best-of-K refinement duplicates each trajectory into $K{=}4$ copies after the split and keeps the highest-reward suffix (3.9), sharing prefix computation. Revealed tokens stay fixed; the practical sampler does not inherit the exact-target guarantee.

time. This guarantee is a statement about the true rollout value and the exact normalized kernel, not the prediction-head implementation. In Section 3, the head proxy, finite candidate set, proposal and completion temperatures, weight sharpening, and terminal selection each have a distinct role. Finite-candidate convergence alone cannot remove a systematic value or proposal mismatch. We keep the original P-referenced π_β as the theoretical benchmark; Section B separately states the exact transform for a temperature-modified reference chain. The practical method in Section 3 estimates this ratio without retraining the masked diffusion model. The continuous analogue appears as an additive score correction $\nabla \log p_t + \nabla \log h_t$. The categorical and irreversible reverse dynamics of masked diffusion have no Euclidean score field, and instead require the multiplicative kernel reweighting in (2.5); Section C records the continuous derivation.

3 D-DOIT: A Discrete Sampler Based on Doob's h-transform

In this section, we introduce **D-DOIT**, a training-free sampler that guides a frozen masked discrete diffusion model toward the reward-tilted target in (2.3) using black-box rewards $R : \mathcal{X} \to \mathbb{R}$ defined on clean sequences. The exact transform in Theorem 2.1 says what the sampler should do: reweight each frozen reverse kernel by the downstream value ratio of the discrete Doob h-function. The challenge is computational, since evaluating this value exactly would require future denoising rollouts from every candidate next state. D-DOIT therefore follows an exact-object-to-estimator design: it estimates the downstream value with the frozen prediction head, approximates the transformed kernel with finite candidates, and optionally branches late to hedge irreversible token reveals. Figure 1 summarizes the construction.

3.1 D-DOIT Algorithm

The transformed kernel in (2.5) depends on the downstream value $h(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1})$ for each candidate reverse state $\bar{\mathbf{y}}_{t_{l-1}} \in \widetilde{\mathcal{X}}$ reachable from the current state $\bar{\mathbf{x}}_{t_l}$. This subsection turns that exact identity

into an executable sampler. The first component estimates each candidate's downstream value with a prediction-head completion. The second component uses these values to form a finite-candidate approximation to the transformed kernel. The final component adds an optional best-of-K refinement for the irreversible reveal structure of masked diffusion.

Prediction-head h-estimator. The exact value $h(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1})$ averages the reward weight over all complete denoising continuations from the candidate next state $\bar{\mathbf{y}}_{t_{l-1}}$. Naive Monte Carlo would roll out the frozen reverse chain many times from every candidate. D-DOIT avoids these nested rollouts by evaluating candidates with the masked discrete diffusion prediction head at $\bar{\mathbf{y}}_{t_{l-1}}$. Recall from Section 2.1 that $\mu_\theta^j(\cdot \mid \mathbf{x}_{t_l}, t_l) \in \Delta(\mathcal{V})$ denotes the neural network's approximation to the clean-token posterior at coordinate j. The head provides a direct clean-sequence completion distribution by fixing revealed tokens and independently sampling still-masked positions from the coordinate-wise clean-token posteriors $\mu_\theta^j(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}}, t_{l-1})$; we write p_θ^{head} for this direct completion law, which is not generally a rollout conditional:

$$p_\theta^{\mathrm{head}}(\mathbf{x}_0 \mid \bar{\mathbf{y}}_{t_{l-1}}) = \prod_{j:\, \bar{y}^j_{t_{l-1}} \neq \mathsf{m}} \delta_{\bar{y}^j_{t_{l-1}}}(x_0^j) \prod_{j:\, \bar{y}^j_{t_{l-1}} = \mathsf{m}} \mu_\theta^j(x_0^j \mid \bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}). \tag{3.1}$$

Definition 3.1 (Prediction-head h-estimator)**.** For every reachable candidate reverse state $\bar{\mathbf{y}}_{t_{l-1}} \in \widetilde{\mathcal{X}}$, D-DOIT estimates the downstream value by the one-step clean-completion expectation under the factorized prediction-head distribution for each candidate state:

$$\bar{h}_\theta(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) = \mathbb{E}_{\mathbf{x}_0 \sim p_\theta^{\mathrm{head}}(\mathbf{x}_0 \mid \bar{\mathbf{y}}_{t_{l-1}})}[\phi(\mathbf{x}_0)]\,. \tag{3.2}$$

The estimator (3.2) replaces the rollout value $h(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1})$ from Definition 2.1 with a single-step completion value. This is the main approximation that makes the transform practical; Section D discusses its bias relative to full future rollouts. For each candidate, we estimate (3.2) with one tempered completion $\hat{\mathbf{x}}_0 \sim p_{\theta,\gamma}^{\mathrm{head}}(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}})$:

$$\hat{h}(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) = \exp(R(\hat{\mathbf{x}}_0)/\beta), \qquad \hat{\mathbf{x}}_0 \sim p_{\theta,\gamma}^{\mathrm{head}}(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}}), \tag{3.3}$$

$$\mu_{\theta,\gamma}^j(v \mid \bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) \propto \mu_\theta^j(v \mid \bar{\mathbf{y}}_{t_{l-1}}, t_{l-1})^{1/\gamma}. \tag{3.4}$$

The completion temperature γ tempers the *completion* of a candidate $\bar{\mathbf{y}}_{t_{l-1}}$ into a clean sequence. Smaller γ sharpens the completion toward the model mode, while larger γ smooths noisy predictions. Because the reward is queried only on the sampled prediction-head completion $\hat{\mathbf{x}}_0$, $R(\cdot)$ need not be differentiable, and no gradients are backpropagated through the reward oracle. Using n_h independent completions per candidate averages

$$\hat{h}_{n_h}(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) = \frac{1}{n_h} \sum_{i=1}^{n_h} \exp\Big(R(\hat{\mathbf{x}}_0^{(i)})/\beta\Big), \qquad \hat{\mathbf{x}}_0^{(i)} \sim p_{\theta,\gamma}^{\mathrm{head}}(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}}), \tag{3.5}$$

which reduces Monte Carlo variance for this head expectation without removing its discrepancy from the rollout value. The reported main configurations use $n_h = 1$.

Finite-candidate discrete h-transform. At reverse step l, D-DOIT first draws M candidate next states $\{\bar{\mathbf{y}}_{t_{l-1}}^{(m)}\}_{m=1}^M$ from a τ-tempered masked discrete diffusion proposal:

$$\mu_{\theta,\tau}^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l) \propto \mu_\theta^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l)^{1/\tau}, \qquad v \in \mathcal{V}. \tag{3.6}$$

Both temperatures act on the same coordinate-wise prediction head μ_θ^j, but they play different roles: τ controls proposal diversity, while γ controls the clean completion used for reward scoring. We draw each candidate $\bar{\mathbf{y}}_{t_{l-1}}^{(m)}$ from the masked discrete diffusion reverse kernel (2.1) with $p_{\theta,\tau}$ in place of p_θ, keeping the schedule factor $(\alpha_{l-1} - \alpha_l)/(1 - \alpha_l)$ and the copy-through rule. All M proposals share one forward pass, and each is completed once via (3.3). The next state $\bar{\mathbf{x}}_{t_{l-1}} = \bar{\mathbf{y}}_{t_{l-1}}^{(m^*)}$ is drawn by sampling the index m^* over the M candidates, each assigned probability

$$w^{(m)} = \frac{\hat{h}(\bar{\mathbf{y}}_{t_{l-1}}^{(m)}, t_{l-1})^\omega}{\sum_{r=1}^M \hat{h}(\bar{\mathbf{y}}_{t_{l-1}}^{(r)}, t_{l-1})^\omega}. \tag{3.7}$$

With $\tau = 1$, (3.7) is a finite-candidate self-normalized approximation to the h-reweighted transition induced by (2.5), with ω controlling how sharply the sampler follows the estimated downstream value. For one completion, $\hat{h}^{\omega} = \exp(\omega R(\hat{\mathbf{x}}_0)/\beta)$, so ω is not a separate approximation. Every (ω, β) pair is equivalent to $\omega = 1$ with effective inverse temperature $\eta = \omega/\beta$, the parameter used in Section B. With $\tau > 1$, the proposal is deliberately smoothed, so the weighted selection rule becomes a biased proposal approximation that we use to maintain candidate diversity. Retaining the original P-referenced target under a smoothed proposal would instead require the importance weight

$$\widetilde{w}^{(m)} \propto \frac{P_{\theta,t_l}(\bar{\mathbf{y}}^{(m)}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l})}{Q^{\tau}_{l}(\bar{\mathbf{y}}^{(m)}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l})} h(\bar{\mathbf{y}}^{(m)}_{t_{l-1}}, t_{l-1}), \tag{3.8}$$

where Q^{τ}_{l} is the proposal kernel, assuming its support covers the target transition. Our operating points omit this ratio, so it is a reference correction, not the executed algorithm. Even with it, a biased proxy, sharpening, and terminal best-of-K selection would remain. A guided step costs $1 + M$ model-side NFE: one shared proposal pass and M completion passes. It also costs M reward evaluations; Section G gives the full accounting and cost conventions.

Best-of-K refinement. Revealed tokens cannot be revised, so D-DOIT optionally duplicates each active trajectory into K copies after split update s_{split}, counted in reverse updates. The copies continue independently, and the method keeps the highest-reward clean sequence within each group. For each original trajectory b, the selected output is

$$\mathbf{x}_0^{(b)} = \mathbf{x}_0^{(b,k_b^*)}, \qquad k_b^* = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} R(\mathbf{x}_0^{(b,k)}). \tag{3.9}$$

The $K\times$ multiplier is paid only after the split, hedging early decisions without always multiplying the full trajectory cost. This terminal argmax changes the output law and cannot repair earlier reveals. Before the split a trajectory is shared; after it the K copies update independently, and with $K = 1$ or $s_{\text{split}} = L$ the method reduces to D-DOIT. Algorithms 2 and 3 in the appendix give the sampler and its per-step guided update.

3.2 Two Configurations: D-DOIT-LBOK-S and D-DOIT-Q

We report two operating points of the same sampler that differ only in budget allocation. **D-DOIT-LBOK-S** uses a small per-step candidate budget plus late best-of-K refinement for the practical lower-budget regime. **D-DOIT-Q** allocates more inference budget to candidate evaluation and/or split-time refinement to test the quality frontier of the frozen backbone. Table 5 lists the benchmark-specific values, and Section 4 evaluates both configurations against training-free and training-based baselines.

4 Experiments

In this section, we include the following results: (1) regulatory DNA enhancer design (Section 4.1); (2) cell-type-specific enhancer design (Section 4.2); (3) protein inverse folding (Section 4.3); (4) contrast with the closest reward-guided sampling baselines (Section 4.4); (5) analysis of the estimation accuracy of $\hat{\mathbf{x}}_0$ (Section 4.5); (6) analysis of the impact of different reward functions (Section 4.6).

Table 1: **Comparison of reward-guided sampling methods for discrete diffusion.** We contrast D-DOIT with our guidance baselines (Section 4); CG = classifier guidance, TDS = twisted diffusion sampler, SMC = sequential Monte Carlo, SVDD = soft value-based decoding.

Property	CG	TDS/SMC	SVDD	D-DOIT (Ours)
Reward intervention	gradient (backward)	particle weights	soft value estimate	Doob h-transform on the discrete kernel
Requires backprop	✓	✗	✗	✗
Weighted selection unit	—	M particle trajectories	M candidate states	M candidate next states (same $\mathbf{x}_l$)
Weight formula	—	path likelihood ratio	estimated soft value	transformed kernel via $\hat{h}^{\omega}$
Doob kernel tilt	✗	✗	✗	✓
Candidate diversity	—	different histories	same state start	same $\mathbf{x}_l$ start

Furthermore, we conducted detailed ablation studies on the D-DOIT algorithm using the regulatory DNA sequence design task to analyze the relationship between its computational efficiency and performance, and compared it with other algorithms (Sections J and L). The task cell type specific enhancer design also features a detailed efficiency-performance analysis (Section K. Detailed

experimental setups are also provided for the protein inverse folding task, serving as a complement to cross-modal transfer (Section M). Table 1 shows the differences between our algorithm and several other algorithms.

4.1 Task 1: Regulatory DNA Sequence Design

Regulatory DNA elements direct cell-type-specific gene expression. They form an important target for gene therapy [24]. We use the HepG2 enhancer design benchmark of [8]: generate 200 bp enhancer sequences with high predicted HepG2 activity and natural enhancer statistics on a frozen masked discrete diffusion backbone. The dataset, reward oracle, checkpoint, protocol, metrics, and baselines are documented in Sections J.1 and J.2.

Table 2: **HepG2 enhancer design results.** We compare D-DOIT with training-free guidance and training-based finetuning baselines on DRAKES HepG2; all metrics are higher-is-better. Parentheses denote standard deviation across seeds, and **bold**/underline mark best/second best per column. D-DOIT uses held-out eval/test seed protocols in Sections J.6 and J.7. †SVDD-PM [12] is the matched-time control from Table 6 ($M = 416$), not part of the original DRAKES sweep.

Method	Pred-Activity (median) ↑	ATAC-Acc ↑ (%)	3-mer Corr ↑	JASPAR Corr ↑
Training-free methods				
Pre-trained	0.17(0.04)	1.5(0.2)	-0.061(0.034)	0.249(0.015)
CG	3.30(0.00)	0.0(0.0)	-0.065(0.001)	0.212(0.035)
SMC	4.15(0.33)	39.9(8.7)	0.840(0.045)	0.756(0.068)
TDS	4.58(0.49)	40.0(14.1)	0.817(0.097)	0.798(0.044)
SVDD-PM†	6.04(0.07)	39.8(0.4)	0.688(0.012)	0.917(0.002)
D-DOIT-LBOK-S (Ours)	4.56(0.04)	65.0(1.4)	0.851(0.017)	**0.948(0.002)**
D-DOIT-Q (Ours)	6.19(0.05)	91.7(0.9)	**0.894(0.009)**	0.923(0.002)
Training-based methods				
CFG	5.04(0.06)	92.1(0.9)	0.746(0.001)	0.864(0.011)
DRAKES w/o KL	**6.44(0.04)**	82.5(2.8)	0.307(0.001)	0.557(0.015)
DRAKES	5.61(0.07)	**92.5(0.6)**	0.887(0.002)	0.911(0.002)

Table 2 reports quality metrics. D-DOIT-Q outperforms every training-free baseline in Table 2 on every metric. D-DOIT-Q reaches 6.19 Pred-Activity, 91.7% ATAC-Acc, 0.894 3-mer Corr, and 0.923 JASPAR Corr. It exceeds training-based DRAKES on Pred-Activity, 3-mer Corr, and JASPAR Corr, while remaining slightly lower on ATAC-Acc (91.7% vs. 92.5%). The near-TDS-speed D-DOIT-LBOK-S matches TDS on Pred-Activity (4.56 vs. 4.58) while lifting ATAC-Acc from 40.0% to 65.0%, 3-mer Corr from 0.817 to 0.851, and JASPAR Corr from 0.798 to 0.948 over TDS. At a similar measured wall-clock budget, Q improves all four metric means over SVDD-PM and improves activity, accessibility, and 3-mer correlation over Frozen Best-of-732; endpoint search retains higher JASPAR correlation. Section J.3 and Table 6 gives the matched-time results and full protocol. A guided step costs $1 + M$ model forward passes plus M reward evaluations. The high-budget Q row is calibrated to endpoint sampling at approximately 10.6 seconds per output on the same A100; Section G gives the accounting.

4.2 Task 2: Cell-type-specific Enhancer Design

Cell-type-specific enhancer design asks for sequences active in a target cell line and silent in off-target cell lines. This task is a stricter test of controllable generation than Section 4.1. We conduct experiments on the Ctrl-DNA enhancer benchmark of [25]. The benchmark spans three cell lines (HepG2, K562, SK-N-SH) and supplies cell-type-specific Enformer-based reward oracles. The dataset, composite specificity reward, per-cell-line oracles, diffusion training details, and the baseline-tuning protocol are documented in Sections K.1 and K.3.

We report specificity (ΔR), Motif Corr, and set-level diversity; ΔR is the target-cell oracle score minus the average off-target score. Baselines include the training-free methods as in Section 4.1, regLM [26], RL baselines [25], and directed evolution; full metric definitions and baseline tuning are documented in the appendix, Sections K.2 and K.3.

Table 3 reports three-cell-line averages. Among the diffusion-guidance baselines, D-DOIT clearly improves specificity: D-DOIT-LBOK-S raises ΔR from 0.002 to 0.033 over TDS while improving

Motif Corr from 0.573 to 0.648 at near-TDS wall-clock. D-DOIT-Q pushes ΔR to 0.119 while trading motif preservation for reward, giving the high-specificity point; LBOK-S remains the lower-budget point. Under the matched-time control, endpoint search also improves specificity and captures much of this gain, unlike the larger activity and accessibility separation on DRAKES. Q's mean differences from Frozen Best-of-708 are small and are not claimed statistically significant, so the specificity gain is not unique to D-DOIT here. Wall-clock, per-cell, and diversity details and the matched-time control are in Sections K.4 and K.5.

4.3 Task 3: Protein Inverse Folding

To test a different sequence space and reward, we apply D-DOIT to the DRAKES protein inverse-folding benchmark [8], using its reward oracle and evaluation metrics. Here the target backbone is fixed and the sampler generates amino-acid sequences. We evaluate predicted stability and agreement between the predicted structure and that backbone, with the full cross-method comparison including every reported metric for this benchmark in Table 3.

Across all 12 test backbones, three seeds, and 128 sequences per backbone and seed, D-DOIT achieves joint success of $68.1 \pm 3.5\%$, compared with the published SMC and TDS results of 63.6% and 62.9% (Table 3). The configuration $M = 52$, $K = 8$, $n_h = 1$ was fixed before test evaluation, with no post-test tuning, and each entry reports the mean and sample SD across the three seed-level metrics. D-DOIT has the highest predicted stability and joint success among the six methods, while TDS has better structural fidelity. The other five rows are taken from DRAKES Table 2, not rerun or compute-matched controls. Section M gives the full protocol. This supports transfer to a second biological sequence-design task, not experimental protein validation.

Table 3: **Cell-type-specific enhancer design (left) and protein inverse folding (right).** Left: means over three cell lines, SD in parentheses; ΔR is target minus average off-target reward; training-based references are in Table 13 and Section K.4. Right: DRAKES protein benchmark; Pred-ddG and joint success are higher-is-better, scRMSD lower-is-better, SDs in Table 22. Bold marks the best mean per column, not significance.

Method	$\Delta R \uparrow$	Motif Corr. ↑
Training-free methods		
Pre-trained	0.000(0.003)	0.937(0.009)
CG	0.000(0.003)	**0.938(0.009)**
SMC	0.000(0.010)	0.587(0.114)
TDS	0.002(0.009)	0.573(0.078)
Frozen Best-of-708	0.108(0.028)	0.490(0.165)
D-DOIT-LBOK-S	0.033(0.025)	0.648(0.046)
D-DOIT-Q	0.119(0.063)	0.498(0.070)

Method	Pred-ddG ↑	scRMSD ↓	Success (%) ↑
Training-free methods			
Pre-trained	−0.544	0.849	34.4
CG	−0.561	0.839	34.7
SMC	0.659	0.841	63.6
TDS	0.674	**0.834**	62.9
D-DOIT (Ours)	**0.870**	0.872	**68.1**
Training-based methods			
CFG	−1.186	3.146	1.3

4.4 Compute–quality Trade-off

To compare compute across methods, we account for oracle reward calls in addition to model evaluations. A guided update evaluates a shared proposal head, M candidate heads, and one reward per completed candidate; branching adds work only after the split.

The lower-budget DRAKES setting uses 1,970 model evaluations and 1,773 candidate completions per output, with 1,777 reward evaluations after ranking the four final branches. Terminal ranking adds up to K queries when scores are not cached. In the 640-output diagnostic, 1,290 vectorized oracle calls score 1,137,280 sequences, so physical calls and scored sequences are distinct cost units. Full accounting is given in Section G.

Table 4: **Per-output DRAKES work.** Full-window guidance, $L = 128$, one completion per candidate.

Quantity	LBOK-S	Q
(M, K, s_{split})	$(9, 4, 105)$	$(52, 8, 0)$
Model NFE	1,970	54,272
Candidate completions	1,773	53,248

Model NFE alone therefore does not determine total compute. In Table 7, Frozen Best-of-15 has a similar model budget (1,920 versus 1,970 evaluations per output) but only 15 rather than 1,777 oracle evaluations. Its 640-output job is faster (97.8 versus 327.8 seconds) but has lower Pred-Activity (2.219 versus 4.491). Thus, this comparison approximately matches

model budget, not wall-clock time. The calibration in Table 6 instead matches endpoint sampling to Q at approximately 10.6 seconds per output.

LBOK-S remains the near-TDS-runtime operating point ($1.16\times$ in the main timing study), whereas Q allocates substantially more compute to quality. For Ctrl-DNA, Q and Frozen Best-of-708 take 5.43 and 5.51 seconds per output on the same A100. Because absolute timings depend on batching and memory use, we report each study under its own timing protocol. D-DOIT generation costs recur for new samples, and comparisons with training-based methods concern sample quality rather than matched or amortized training cost.

4.5 Completion Quality and Search Diagnostics

A crucial aspect of estimating h is how to obtain $\hat{\mathbf{x}}_0$; while a common approach to maintaining speed is to estimate $\hat{\mathbf{x}}_0$ directly from x_t in a single step, a more refined method involves obtaining $\hat{\mathbf{x}}_0$ through iterative steps starting from x_t. To better compare these two approaches, we investigated the performance of different estimation methods. Increasing completions per candidate from $n_h = 1$ to 2 and 3 raises final guidance reward R from 5.272 ± 0.057 to 5.764 ± 0.033 and 5.978 ± 0.026, at 41% and 81% additional wall time; we retain $n_h = 1$ as the default. Direct and sequential completions have similar h-value distributions (Wasserstein-1 0.060–0.124; Section L.5), supporting practical use without exact-target equivalence.

4.6 Oracle Robustness

The choice of reward function is independent of the D-DOIT algorithm itself, yet it significantly influences the algorithm's performance. A more accurate reward function provides better guidance for the D-DOIT algorithm. To examine this, we designed various reward functions and observed their guiding effects. Replacing R with Enformer, CNN, or MLP predictions improves all four metrics over unguided sampling (Table 8); Pred-Activity reaches 4.491, 2.203, and 1.544, versus 0.371 for the frozen model. Held-out rescoring preserves these gains (Table 9), while evaluator scales remain incomparable. The controls support robustness across tested predictors without establishing experimental biological activity and are distinct from the high-budget 6.19 ± 0.05 result.

5 Conclusion and Discussion

We introduced D-DOIT, a training-free sampler that applies Doob's h-transform directly to frozen masked discrete diffusion models. Using a prediction-head h-estimator and finite-candidate Doob-weighted selection, D-DOIT supports black-box rewards without reward gradients or backbone retraining. Across DRAKES HepG2, Ctrl-DNA, and protein inverse folding, D-DOIT outperforms training-free guidance baselines, although its highest-quality results require a larger compute budget.

The method has three main limitations. First, practical approximations, including temperature scaling, sharpening, and best-of-K selection, may introduce bias and do not retain the exact-target guarantee; late branching also cannot revise revealed tokens. Second, evaluation relies on predictive biological benchmarks and requires experimental validation. Third, the highest-quality setting incurs substantial recurring inference cost, leaving the reward–compute trade-off as a practical challenge. Overall, Doob-style kernel reweighting provides a general framework for guiding discrete generative models with sequence-level rewards.

References

[1] 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, 2021. doi: 10.48550/arxiv.2107.03006.

[2] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35, 2022. doi: 10.52202/068431-2049.

[3] 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*, 2024.

[4] 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 (NeurIPS)*, 2024.

[5] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arxiv.2310.16834.

[6] Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arxiv.2402.05841.

[7] Anirban Sarkar, Alejandra Duran, Yiyang Yu, Da-Wei Lin, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Zhou, Masayuki Nagai, Ziqi Tang, Kaarina Hanington, Kenneth Chang, and Peter K. Koo. Designing DNA with tunable regulatory activity using discrete diffusion. *bioRxiv*, 2024. doi: 10.1101/2024.05.23.595630.

[8] 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 (ICLR)*, 2025. doi: 10.48550/arxiv.2410.13643.

[9] 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. *arXiv preprint arXiv:2507.04832*, 2025.

[10] 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 (ICLR)*, 2025. doi: 10.48550/arxiv.2406.01572.

[11] Luhuan Wu, Brian A Trippe, Christian A Naesseth, David M Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[12] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen 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*, 2025. doi: 10.48550/arxiv.2408.08252.

[13] 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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 60515–60529, 2025. doi: 10.48550/arxiv.2502.14944.

[14] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-Torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arxiv.2412.10193.

[15] Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Chenghao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arxiv.2410.08134.

[16] Wei Guo, Yuchen Zhu, Molei Tao, and Yongxin Chen. Plug-and-play controllable generation for discrete masked models. *arXiv preprint arXiv:2410.02143*, 2024. doi: 10.48550/arxiv.2410.02143.

[17] 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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 55810–55827, 2025.

[18] 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 Computer Vision and Pattern Recognition Conference*, pages 28490–28501, 2025. doi: 10.1109/cvpr52734.2025.02653.

[19] 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. doi: 10.48550/arxiv.2602.16198.

[20] Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching. *Biometrika*, 112(4):asaf048, 2025. doi: 10.1093/biomet/asaf048.

[21] Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon V. Mathis, Riccardo Barbano, Vincent Dutordoir, 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 (NeurIPS)*, 2024. doi: 10.48550/arxiv.2406.01781.

[22] L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000. doi: 10.1017/cbo9780511805141.

[23] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019. doi: 10.1017/9781108186735.

[24] 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.

[25] Xingyu Chen, Shihao Ma, Runsheng Lin, Jiecong Lin, and Bo Wang. Ctrl-DNA: Controllable cell-type-specific regulatory DNA design via constrained RL. *arXiv preprint arXiv:2505.20578*, 2025. doi: 10.48550/arxiv.2505.20578.

[26] Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. RegLM: Designing realistic regulatory DNA with autoregressive language models. *bioRxiv*, pages 2024–02, 2024. doi: 10.1101/2024.02.14.580373.

[27] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[28] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[29] Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Qun Liu, Chen Luo, Jiri Gesi, Hanqing Lu, Yisi Sang, Manling Li, Jing Huang, et al. Sentinel: Failure-driven reinforcement learning for training tool-using language model agents. *arXiv preprint arXiv:2606.12908*, 2026.

[30] Ziyi Wang, Yuxuan Lu, Wenbo Li, Amirali Amini, Bo Sun, Yakov Bart, Weimin Lyu, Jiri Gesi, Tian Wang, Jing Huang, et al. Opera: A dataset of observation, persona, rationale, and action for evaluating llms on human online shopping behavior simulation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 43942–43960, 2026.

[31] Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Jing Huang, and Dakuo Wang. Customer-r1: Personalized simulation of human behaviors via rl-based llm agent in online shopping. *arXiv preprint arXiv:2510.07230*, 2025.

[32] Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Pei Chen, Ziwei Dong, Jing Huang, Jiri Gesi, Xianfeng Tang, Chen Luo, Qun Liu, et al. Trajectory2task: Training robust tool-calling agents with synthesized yet verifiable data for complex user intents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44021–44044, 2026.

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

[34] Jerry Yao-Chieh Hu, Weimin Wu, Zhuoru Li, Sophia Pi, Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (dits). *Advances in Neural Information Processing Systems*, 37:31562–31628, 2024.

[35] Jerry Yao-Chieh Hu, Weimin Wu, Yi-Chen Lee, Yu-Chao Huang, Minshuo Chen, and Han Liu. On statistical rates of conditional diffusion transformers: Approximation, estimation and minimax optimality. In *International Conference on Learning Representations*, volume 2025, pages 12591–12675, 2025.

[36] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. doi: 10.48550/arxiv.2102.05379.

[37] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arxiv.2406.03736.

[38] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arxiv.2402.04997.

[39] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[40] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arxiv.2305.13301.

[41] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[42] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arxiv.2309.17400.

[43] 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 (ICLR)*, 2025. doi: 10.48550/arxiv.2409.08861.

[44] Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arxiv.2206.04119.

[45] Gabriel Cardoso, Sylvain Le Corff, and Eric Moulines. Monte carlo guided diffusion for bayesian linear inverse problems. In *International Conference on Learning Representations (ICLR)*, 2024.

[46] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arxiv.2209.14687.

[47] Prin Phunyaphibarn and Minhyuk Sung. Reward-guided discrete diffusion via clean-sample markov chain for molecule and biological sequence design. *arXiv preprint arXiv:2602.09424*, 2026. doi: 10.48550/arXiv.2602.09424. URL https://arxiv.org/abs/2602.09424.

[48] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025. doi: 10.48550/arxiv.2501.09732.

[49] Joseph L Doob. *Classical Potential Theory and Its Probabilistic Counterpart*. Springer, Berlin, 1984. doi: 10.1007/978-3-642-56573-1.

[50] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[51] Jaihoon Kim, Taehoon Yoon, Prin Phunyaphibarn, Seungjun Kim, Morteza Mardani, and Minhyuk Sung. Contrastive distribution matching for amortized sequential monte carlo in discrete diffusion. *arXiv preprint arXiv:2605.23346*, 2026. URL https://arxiv.org/abs/2605.23346.

[52] Dake Bu, Wei Huang, Andi Han, Si Wu, Hau-San Wong, Qingfu Zhang, Taiji Suzuki, and Atsushi Nitanda. DPRM: A plug-in doob h transform-induced token-ordering module for discrete diffusion models. *arXiv preprint arXiv:2604.24357*, 2026. URL https://arxiv.org/abs/2604.24357.

[53] Ibrahim I Taskiran, Katina I Spanier, Hannah Dickmanken, Niklas Kempynck, Alexandra Pančíková, Eren Can Eksşi, Gert Hulselmans, Joy N Ismail, Koen Theunis, Roel Vandepoel, et al. Cell-type-directed design of synthetic enhancers. *Nature*, 626(7997):212–220, 2024. doi: 10.1038/s41586-023-06936-2.

[54] Maojiang Su, Po-Chung Hsieh, Weimin Wu, Mingcheng Lu, Jiunhau Chen, Jerry Yao-Chieh Hu, and Han Liu. Discrete flow matching policy optimization. *arXiv preprint arXiv:2604.06491*, 2026.

[55] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, et al. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in Neural Information Processing Systems*, 2023.

[56] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arxiv.2403.03234.

[57] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, 2024. doi: 10.1038/s41592-024-02523-z.

[58] Zhihan Zhou, Robert Riley, Satria Kautsar, Weimin Wu, Rob Egan, Steven Hofmeyr, Shira Goldhaber-Gordon, Mutian Yu, Harrison Ho, Fengchen Liu, et al. Genomeocean: An efficient genome foundation model trained on large-scale metagenomic assemblies. *bioRxiv*, 2025.

[59] Weimin Wu, Xuefeng Song, Yibo Wen, Qinjie Lin, Zhihan Zhou, Jerry Yao-Chieh Hu, Zhong Wang, and Han Liu. Genome-factory: A library for tuning, deploying, and interpreting genomic foundation models. *arXiv preprint arXiv:2509.12266*, 2025.

[60] Zhihan Zhou, Weimin Wu, Harrison Ho, Jiayi Wang, Lizhen Shi, Ramana V Davuluri, Zhong Wang, and Han Liu. Dnabert-s: pioneering species differentiation with species-aware dna embeddings. *Bioinformatics*, 41(Supplement_1):i255–i264, 2025.

[61] Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning (ICML)*, 2023. doi: 10.48550/arxiv.2305.10699.

[62] Animesh Awasthi, Raphael Bednarsky, Moritz Schaefer, and Christoph Bock. Conditional monte carlo tree diffusion for designing cell-type-specific and biologically faithful regulatory DNA. *arXiv preprint arXiv:2604.20488*, 2026. doi: 10.48550/arxiv.2604.20488.

[63] Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. doi: 10.1016/0304-4149(82)90051-5.

[64] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, 2021. doi: 10.1038/s41592-021-01252-x.

[65] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. doi: 10.1038/nature11247.

[66] Jaime A Castro-Mondragón, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, et al. 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.

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

Appendix

A Related Work 17

B Proof of Theorem 2.1 18

C Continuous-time Doob's h-transform on Score-based Diffusion 20

D On the Prediction-head h-estimator 21

E Standard Masked Discrete Diffusion Sampling 23

F Masked Diffusion Transition Details 24

G Per-output NFE Breakdown 25

H Detailed Sampler Pseudocode 26

I Additional Analyses 27

J Experimental Details of Regulatory DNA Sequence Design 28

J.1 Dataset and Training Details . . . 28
J.2 Evaluation Metrics and Baselines . . . 28
J.3 Nearest-neighbor and Endpoint-search Controls . . . 29
J.4 Guidance-oracle and Evaluator Robustness . . . 30
J.5 Additional Experimental Analysis . . . 30
J.6 Held-out Final-test Check for D-DOIT-LBOK-S . . . 31
J.7 Held-out Selection for High-budget D-DOIT-Q . . . 31

K Experimental Details of Cell-type-specific Enhancer Design 33

K.1 Dataset and Training Details . . . 33
K.2 Evaluation Metrics and Baselines . . . 33
K.3 Training-free Baseline Tuning Screen . . . 34
K.4 Additional Experimental Analysis . . . 34
K.5 Wall-clock-matched Endpoint Control . . . 35

L Ablation Study 37

L.1 Effect of Guidance Window l^* . . . 37
L.2 Effect of Guidance Strength ω and ESS Collapse . . . 38
L.3 Effect of Proposal and Completion Temperatures τ, γ . . . 38
L.4 Effect of Candidate Budget M . . . 38
L.5 Completion Count and Matched-state Proxy Diagnostics . . . 39
L.6 Soft Selection versus Argmax . . . 40
L.7 Split Time and Reward Trajectories . . . 40

M Experimental Details of Protein Inverse Folding 42

N Licenses for Existing Assets 43

A Related Work

We review related work on reward-guided adaptation of discrete diffusion and flow models, as well as regulatory DNA sequence design.

Reward-guided adaptation of discrete diffusion and flow models. Reward-guided generation has proved effective for aligning language models [27–32] and diffusion models [33–35] with task-specific objectives. In discrete generative models, however, guidance must act on categorical transitions over a finite vocabulary. Early discrete denoising diffusion models use structured categorical or absorbing-state corruption [1, 36, 2]; masked discrete diffusion models scale a simple absorbing-mask objective [3, 4]; score-entropy and ratio-matching parameterizations form a non-masked alternative [5, 37]; and discrete flow matching extends the same idea to a flow framework [38, 39]. Reward adaptation for these models can be training-based or training-free. Training-based methods include DRAKES, which finetunes the generator through reward backpropagation [8], trajectory-level alignment [9], and reinforcement learning (RL)-style finetuning methods developed for continuous diffusion [40–43]; these methods change backbone weights and need a differentiable reward. Training-free methods instead steer a frozen backbone at inference time, including Sequential Monte Carlo and twisted sampling [11, 44, 45], posterior sampling for inverse problems [46], soft value-based decoding [12], reward-guided iterative refinement [13], clean-sample Markov chains [47], simple guidance for masked discrete diffusion [14], denoising posterior prediction [15], plug-and-play control [16], and inference-time scaling [17, 48]. These methods steer a frozen backbone, but they do not directly tilt the discrete reverse kernel toward an optimal proposal.

Comparing with three representative efforts highlights the distinct role of D-DOIT. DRAKES [8] provides a strong reward-optimization baseline, but it finetunes the backbone and requires a differentiable reward. Soft value-based decoding [12] selects candidates using importance scores on intermediate latents, but it does not tilt the reverse transition kernel itself. Simple masked-diffusion guidance [14] introduces a heuristic guidance term, but lacks an explicit optimality target. In contrast, D-DOIT is the first inference-time method to implement an executable estimator of Doob's h-transform for the discrete reverse kernel [49, 19–21, 50]. It scores candidates through the resulting h-tilted kernel, combines a frozen masked discrete diffusion backbone with a black-box reward, and further supports optional best-of-K refinement over the same frozen prior. Because finite-candidate value-weighted selection also appears in soft value-based decoding and related methods, we do not claim that local candidate selection is unique to D-DOIT. CDM [51] learns a twist to amortize value estimation, and DPRM [52] applies Doob-derived corrections to token ordering while retaining a host token sampler; CSMC [47] instead searches over clean samples. Our distinction is the explicit categorical reverse-kernel h-transform as the reference object, evaluated with a frozen prediction-head proxy rather than a learned value network.

Regulatory DNA sequence design. Regulatory DNA design aims to generate sequences that score well under predictive assays and stay faithful to natural regulatory grammar. Oracle-guided optimization with experimental validation drives early machine-guided enhancer design [53, 24, 54]; autoregressive regulatory language models [26] and genomic foundation models [55–60] provide strong sequence priors; and diffusion-based generators specialize these priors through Dirichlet diffusion and score-entropy variants [6, 7, 61]. For reward-optimization benchmarks, DRAKES evaluates HepG2 enhancer generation through activity, accessibility, k-mer, and motif-structure metrics [8]; Ctrl-DNA studies cell-type-specific enhancer design through constrained reinforcement learning [25]; and concurrent work combines discrete diffusion with tree search for cell-type-specific regulatory DNA design under related biological reward objectives [62]. The strongest methods on these benchmarks share one limit: they reach high reward only through backbone updates, policy training, or expanded inference search.

Comparing with three representative efforts highlights the efficiency and generality of D-DOIT. DRAKES [8] finetunes the backbone and requires a differentiable reward. Ctrl-DNA [25] achieves cell-type specificity through constrained RL with reward signals shaped for policy optimization. The concurrent tree-search method of Awasthi et al. [62] remains training-free at inference time, but expands a search tree over partial sequences and increases the per-step sampling cost. In contrast, D-DOIT is the first method to compete on both benchmarks without policy training. It trades modest inference-time compute for reward gains while preserving near-base per-step cost.

B Proof of Theorem 2.1

Proof. Let

$$\phi(\mathbf{x}_0) = \exp(R(\mathbf{x}_0)/\beta), \qquad Z_\beta = \mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0)\right].$$

All identities below are understood on the support of the pretrained reverse chain. Let $\mathbb{P}_\theta$ denote the path law of the pretrained reverse chain,

$$\mathbb{P}_\theta(\bar{\mathbf{x}}_{t_0}, \dots, \bar{\mathbf{x}}_{t_L}) = p_{\theta,t_L}(\bar{\mathbf{x}}_{t_L}) \prod_{l=1}^{L} P_{\theta,t_l}(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}).$$

Define the reward-tilted path law by

$$\frac{\mathrm{d}\mathbb{P}_\theta^h}{\mathrm{d}\mathbb{P}_\theta}(\bar{\mathbf{X}}_{t_0}, \dots, \bar{\mathbf{X}}_{t_L}) = \frac{\phi(\mathbf{X}_0)}{Z_\beta}.$$

We now compute the one-step reverse transition under this tilted path law. For $l = 1, \dots, L$,

$$\begin{aligned}
&\mathbb{P}_\theta^h\left(\bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right) \\
&= \frac{\mathbb{E}_{p_\theta}\left[\mathbf{1}\{\bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{x}}_{t_{l-1}}\}\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right]}{\mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right]} \\
&= P_{\theta,t_l}\left(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}\right) \frac{\mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{x}}_{t_{l-1}}, \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right]}{h(\bar{\mathbf{x}}_{t_l}, t_l)}.
\end{aligned}$$

By the Markov property of the reverse chain, once $\bar{\mathbf{X}}_{t_{l-1}}$ is given, the future evolution toward clean time is independent of $\bar{\mathbf{X}}_{t_l}$. Hence

$$\mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{x}}_{t_{l-1}}, \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right] = h(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1}).$$

Therefore

$$\mathbb{P}_\theta^h\left(\bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right) = P_{\theta,t_l}\left(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}\right) \frac{h(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1})}{h(\bar{\mathbf{x}}_{t_l}, t_l)}.$$

This proves that the reward-tilted path law induces the Doob-transformed reverse kernel in (2.5). In particular, the kernel is normalized because

$$\sum_{\bar{\mathbf{x}}_{t_{l-1}}} P_{\theta,t_l}\left(\bar{\mathbf{x}}_{t_{l-1}} \mid \bar{\mathbf{x}}_{t_l}\right) \frac{h(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1})}{h(\bar{\mathbf{x}}_{t_l}, t_l)} = \frac{\mathbb{E}_{p_\theta}\left[h(\bar{\mathbf{X}}_{t_{l-1}}, t_{l-1}) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right]}{h(\bar{\mathbf{x}}_{t_l}, t_l)} = 1,$$

where the last equality follows from the tower property and the definition of h.

It remains to compute the marginals. For any $l = 0, \dots, L$,

$$\begin{aligned}
p_{\theta,t_l}^h(\bar{\mathbf{x}}_{t_l}) &= \mathbb{P}_\theta^h(\bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}) \\
&= \frac{\mathbb{E}_{p_\theta}\left[\mathbf{1}\{\bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\}\phi(\mathbf{X}_0)\right]}{Z_\beta} \\
&= \frac{p_{\theta,t_l}(\bar{\mathbf{x}}_{t_l})\mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}_{t_l}\right]}{Z_\beta} \\
&= \frac{p_{\theta,t_l}(\bar{\mathbf{x}}_{t_l})h(\bar{\mathbf{x}}_{t_l}, t_l)}{Z_\beta}.
\end{aligned}$$

In the fully masked terminal setting, $\bar{\mathbf{X}}_{t_L}$ is deterministic. Hence

$$h(\bar{\mathbf{X}}_{t_L}, t_L) = \mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0)\right] = Z_\beta,$$

so the transformed terminal distribution coincides with the pretrained terminal distribution.

Finally, since $\bar{\mathbf{X}}_{t_0} = \mathbf{X}_0$,

$$h(\mathbf{x}_0, 0) = \mathbb{E}_{p_\theta}\left[\phi(\mathbf{X}_0) \mid \mathbf{X}_0 = \mathbf{x}_0\right] = \phi(\mathbf{x}_0) = \exp(R(\mathbf{x}_0)/\beta).$$

Therefore

$$p_{\theta,0}^h(\mathbf{x}_0) = \frac{p_{\theta,0}(\mathbf{x}_0)\exp(R(\mathbf{x}_0)/\beta)}{Z_\beta} = \pi_\beta(\mathbf{x}_0).$$

This completes the proof. □

A separate tempered reference chain. The theorem above retains the original frozen P_θ reference and target π_β. The following proposition describes a different reference law; it neither replaces that target nor identifies the executed estimator with an exact Doob value.

Proposition B.1 (Doob h-transform relative to a tempered reference)**.** Fix $\tau > 0$. Let Q_l^τ be the reverse kernel obtained from P_{θ,t_l} by replacing μ_θ with the normalized $\mu_{\theta,\tau}$ in (3.6), while preserving the reveal schedule and copy rule. Let $\mathbb{Q}^\tau$ be the path law using these kernels at every reverse step and starting from the deterministic all-mask state $\bar{\mathbf{x}}_{t_L} = \mathsf{m}^N$. Assume the chain ends in $\mathcal{X}$, and let q_0^τ be its clean-time law. For $\eta > 0$, suppose $0 < Z_{\tau,\eta} := \mathbb{E}_{\mathbb{Q}^\tau}[\exp(\eta R(\mathbf{X}_0))] < \infty$. Define the true reference-chain value, on reachable states, by

$$h_{\tau,\eta}(\bar{\mathbf{x}}, t_l) = \mathbb{E}_{\mathbb{Q}^\tau}[\exp(\eta R(\mathbf{X}_0)) \mid \bar{\mathbf{X}}_{t_l} = \bar{\mathbf{x}}].$$

Then the kernels

$$Q_l^{\tau,h}(\bar{\mathbf{y}} \mid \bar{\mathbf{x}}) = Q_l^\tau(\bar{\mathbf{y}} \mid \bar{\mathbf{x}}) \frac{h_{\tau,\eta}(\bar{\mathbf{y}}, t_{l-1})}{h_{\tau,\eta}(\bar{\mathbf{x}}, t_l)}$$

are normalized. Starting from the same deterministic mask, their clean-time marginal is

$$q_0^{\tau,h}(\mathbf{x}_0) = \frac{q_0^\tau(\mathbf{x}_0) \exp(\eta R(\mathbf{x}_0))}{Z_{\tau,\eta}}.$$

Proof. The Markov and tower properties give

$$h_{\tau,\eta}(\bar{\mathbf{x}}, t_l) = \sum_{\bar{\mathbf{y}}} Q_l^\tau(\bar{\mathbf{y}} \mid \bar{\mathbf{x}}) h_{\tau,\eta}(\bar{\mathbf{y}}, t_{l-1}),$$

which proves normalization and positivity on the reference support. At clean time $h_{\tau,\eta}(\mathbf{x}_0, 0) = \exp(\eta R(\mathbf{x}_0))$; at the deterministic initial mask $h_{\tau,\eta}(\mathsf{m}^N, t_L) = Z_{\tau,\eta}$. Consequently, the likelihood ratio of the transformed path law to $\mathbb{Q}^\tau$ telescopes to

$$\prod_{l=1}^{L} \frac{h_{\tau,\eta}(\bar{\mathbf{x}}_{t_{l-1}}, t_{l-1})}{h_{\tau,\eta}(\bar{\mathbf{x}}_{t_l}, t_l)} = \frac{\exp(\eta R(\mathbf{x}_0))}{Z_{\tau,\eta}}.$$

Marginalizing this identity over the intermediate states yields the claimed clean-time law. □

For $\tau = 1$, $\eta = 1/\beta$ this recovers the P-reference theorem. Otherwise the target changes; the practical proxy, windowing, and terminal selection do not inherit its exactness.

C Continuous-time Doob's h-transform on Score-based Diffusion

We record the standard continuous-time derivation for comparison with the discrete construction. Consider a forward score-based diffusion on $\mathbb{R}^d$,

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{W}_t, \qquad t \in [0, T],$$

with marginal density p_t and $\mathbf{X}_0 \sim p_0$. The reverse-time process admits the well-known form

$$\mathrm{d}\mathbf{X}_t = \big(f(\mathbf{X}_t, t) - g(t)^2 \nabla \log p_t(\mathbf{X}_t)\big)\,\mathrm{d}t + g(t)\,\mathrm{d}\overline{\mathbf{W}}_t,$$

where $\overline{\mathbf{W}}_t$ is reverse-time Brownian motion [63]. For the continuous analogue of the reward-tilted target in (2.3), write the base terminal density as p_0 and denote the target by $\pi(\mathbf{x}_0) \propto p_0(\mathbf{x}_0)\phi(\mathbf{x}_0)$. Define the h-function

$$h_t(\mathbf{x}) = \mathbb{E}\big[\phi(\mathbf{X}_0) \,\big|\, \mathbf{X}_t = \mathbf{x}\big].$$

Doob's h-transform replaces p_t with the tilted marginal $\tilde{p}_t \propto p_t h_t$, which gives the tilted reverse drift

$$\big(f(\mathbf{x}, t) - g(t)^2 \nabla \log p_t(\mathbf{x})\big) - g(t)^2 \nabla \log h_t(\mathbf{x}) = f(\mathbf{x}, t) - g(t)^2 \nabla \log\big(p_t(\mathbf{x}) h_t(\mathbf{x})\big).$$

Equivalently, the score is shifted by an additive correction:

$$\nabla \log p_t(\mathbf{x}) \;\longrightarrow\; \nabla \log p_t(\mathbf{x}) + \nabla \log h_t(\mathbf{x}).$$

With initialization from the tilted marginal at time T, the terminal law of the tilted reverse process is π [19–21]; using the base marginal at T is exact only when that marginal is unchanged by the tilt. The continuous construction requires sufficient regularity of h_t for $\nabla \log h_t$ to exist, not differentiability of R itself. Under suitable smoothing and integrability conditions, differentiating a transition density can give this correction even for a black-box reward. The structural distinction for masked discrete diffusion is instead that $\widetilde{\mathcal{X}}$ is categorical and has no Euclidean score field. Section 3 therefore formulates a discrete kernel-level analogue, where the tilt enters as a multiplicative reweighting of categorical transition probabilities.

D On the Prediction-head h-estimator

Definition 3.1 is the executable counterpart of the rollout h-function used by the exact discrete transform. For attention-based masked discrete diffusion models, it is generally an estimator rather than an exact identity. For a candidate state $\bar{\mathbf{y}}_{t_{l-1}}$ at time t_{l-1}, the exact quantity is

$$h(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) = \mathbb{E}_{p_\theta}[\phi(\mathbf{X}_0) \mid \bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{y}}_{t_{l-1}}],$$

where the expectation runs over the pre-trained reverse chain from time t_{l-1} down to clean time, conditional on $\bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{y}}_{t_{l-1}}$. At each subsequent reverse step $l' \in \{l-1, l-2, \dots, 1\}$, the reverse kernel (2.1) reveals each still-masked position from $\mu_\theta^j(x_0^j \mid \bar{\mathbf{x}}_{t_{l'}}, t_{l'})$ and treats positions as conditionally independent. The copy flag freezes any position that the chain has already revealed. The exact rollout marginal of $\mathbf{X}_0$ given $\bar{\mathbf{X}}_{t_{l-1}} = \bar{\mathbf{y}}_{t_{l-1}}$ therefore mixes per-position predictions at the *intermediate* states $\bar{\mathbf{x}}_{t_{l'}}$ that the chain visits, which carry more revealed context than $\bar{\mathbf{y}}_{t_{l-1}}$.

The factorized head (3.1) evaluates the clean completion from the current state $\bar{\mathbf{y}}_{t_{l-1}}$ itself, giving

$$p_\theta^{\text{head}}(\mathbf{x}_0 \mid \bar{\mathbf{y}}_{t_{l-1}}) = \prod_{j:\, \bar{y}^j_{t_{l-1}} = \mathsf{m}} \mu_\theta^j(x_0^j \mid \bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) \cdot \prod_{j:\, \bar{y}^j_{t_{l-1}} \neq \mathsf{m}} \delta_{\bar{y}^j_{t_{l-1}}}(x_0^j). \tag{D.1}$$

A sufficient condition for this distribution to coincide with the full rollout marginal is that each still-masked coordinate's head prediction remains invariant across all reachable subsequent contexts and times until that coordinate is revealed. For attention-based masked discrete diffusion models this invariance need not hold. The direct completion law therefore generally differs from the multi-step rollout law, even with arbitrarily many completion samples. To separate sampling noise from this distributional mismatch, let n_h denote the number of independent completions of a fixed candidate (reserving N for sequence length), as in (3.5), and write

$$\hat{h}_{n_h}(\bar{\mathbf{y}}, t) = \frac{1}{n_h} \sum_{i=1}^{n_h} \exp\Big(R(\hat{\mathbf{x}}_0^{(i)})/\beta\Big), \qquad \hat{\mathbf{x}}_0^{(i)} \sim p_{\theta,\gamma}^{\text{head}}(\cdot \mid \bar{\mathbf{y}}),$$
$$\bar{h}_{\theta,\gamma}(\bar{\mathbf{y}}, t) = \mathbb{E}_{p_{\theta,\gamma}^{\text{head}}(\cdot \mid \bar{\mathbf{y}})}[\exp(R(\mathbf{X}_0)/\beta)].$$

The executed single-completion estimator has $n_h = 1$. As $n_h \to \infty$, the sample average converges to $\bar{h}_{\theta,\gamma}$, not generally to the rollout value h; at $\gamma = 1$ it recovers the head expectation in (3.2).

Remark D.1 (Practical tempering and selection bias)**.** Completion temperature $\gamma \neq 1$ changes the direct completion law and hence its reward expectation; it does not merely reduce Monte Carlo noise. Proposal temperature $\tau \neq 1$ changes the candidate kernel, denoted Q_l^τ, relative to P_{θ,t_l}. With independent candidate draws and completion randomness, the large-M limit of the uncorrected selection rule at a fixed current state $\bar{\mathbf{x}}$ is proportional to

$$Q_l^\tau(\bar{\mathbf{y}} \mid \bar{\mathbf{x}})\, \mathbb{E}[\hat{h}_{n_h}(\bar{\mathbf{y}}, t_{l-1})^\omega \mid \bar{\mathbf{y}}].$$

On the proposal support, multiplying each weight by $P_{\theta,t_l}(\bar{\mathbf{y}} \mid \bar{\mathbf{x}})/Q_l^\tau(\bar{\mathbf{y}} \mid \bar{\mathbf{x}})$ as in (3.8) corrects only this proposal mismatch, provided the proposal covers the base-kernel support. It does not correct the head-versus-rollout mismatch, completion tempering, nonlinear sharpening, or finite-M self-normalization. The executed sampler omits this likelihood ratio; its uncorrected tempered update is not the exact P_θ transform.

In particular, $\mathbb{E}[\hat{h}_{n_h}^\omega \mid \bar{\mathbf{y}}]$ generally differs from $(\mathbb{E}[\hat{h}_{n_h} \mid \bar{\mathbf{y}}])^\omega$ and from $h(\bar{\mathbf{y}}, t)^\omega$. Even with exact rollout values, raising $\mathbb{E}_P[\exp(R/\beta) \mid \bar{\mathbf{y}}]$ to ω is generally not the same as computing $\mathbb{E}_P[\exp(\omega R/\beta) \mid \bar{\mathbf{y}}]$. For $n_h = 1$, sharpening does yield an expectation of $\exp(\omega R/\beta)$ under the direct completion law, but that law is still not the rollout law. A separate exact transform relative to a tempered reference chain is stated in Section B; it requires that chain's true downstream expectation, not the practical estimator. Finally, restricting guidance to a window omits exact transformed updates outside that window, and terminal best-of-K ranking generally changes the output law even if each branch were sampled exactly. Thus these operating choices are not a guarantee of sampling (2.3); their empirical effects are studied in Section L.3.

Remark D.2 (Comparison with the continuous case)**.** The continuous-time h-function identity in [19] differentiates the Gaussian transition density to bypass the requirement that ϕ be differentiable. The discrete construction avoids differentiating through ϕ for a different reason: we query ϕ only on a

sampled clean completion $\hat{\mathbf{x}}_0 \sim p^{\mathrm{head}}_{\theta,\gamma}(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}})$, so a single forward pass through the prediction head suffices to produce $\hat{h}(\bar{\mathbf{y}}_{t_{l-1}}, t_{l-1}) = \phi(\hat{\mathbf{x}}_0)$ as in (3.3). The trade-off is that this single-shot estimator uses the factorized prediction head (3.1) instead of explicitly simulating the multi-step rollout marginal under the frozen reverse chain.

E Standard Masked Discrete Diffusion Sampling

Algorithm 1 records the standard generative process of the pre-trained masked discrete diffusion model [3] that D-DOIT builds on, provided here as background. The forward process corrupts tokens with an absorbing m state. The reverse process evaluates the clean-sequence prediction head $p_\theta^{\mathrm{head}}(\mathbf{x}_0 \mid \bar{\mathbf{x}}_{t_l})$ at each step. It then unmasks each still-masked position with the per-step reveal schedule from (2.1) and treats positions as conditionally independent. Once a step reveals a token, later steps freeze it. For consistency with the main text, Algorithm 1 writes a linear schedule with $\alpha_{t_l} = 1 - l/L$; other monotone schedules can be substituted.

Algorithm 1 Standard Masked Discrete Diffusion Sampling (Baseline)

Require: Pre-trained masked discrete diffusion model p_θ, steps L

Initialize $\bar{\mathbf{x}}_{t_L} \leftarrow (\mathsf{m}, \ldots, \mathsf{m})$ {all positions masked}
for $l = L, L-1, \ldots, 1$ **do**
  Set $\alpha_{t_l} = 1 - l/L,\ \alpha_{t_{l-1}} = 1 - (l-1)/L$ {linear schedule for this baseline}
  Forward pass: obtain $\mu_\theta^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l)$ for all positions j {one model forward pass}
  for each masked position j (where $\bar{x}_{t_l}^j = \mathsf{m}$) **do**
    Sample $\bar{x}_{t_{l-1}}^j$: unmask to token $v \in \mathcal{V}$ with prob $\mu_\theta^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l)(\alpha_{t_{l-1}} - \alpha_{t_l})/(1 - \alpha_{t_l})$; stay masked with prob $(1 - \alpha_{t_{l-1}})/(1 - \alpha_{t_l})$ {(2.1)}
  end for
  Copy flag: $\bar{x}_{t_{l-1}}^j \leftarrow \bar{x}_{t_l}^j$ for all unmasked j {revealed tokens are frozen}
end for
return $\bar{\mathbf{x}}_0$

F Masked Diffusion Transition Details

This section records the explicit matrices behind the forward and reverse kernels summarized in Section 2.1. For $x \in \widetilde{\mathcal{V}}$, let $e_x \in \mathbb{R}^{d+1}$ be its one-hot vector including e_{m} for the mask token, and let $\mathbf{1} \in \mathbb{R}^{d+1}$ be the all-one vector. Let $(X_{t_l})_{l=0}^{L}$ be the single-token chain with $X_0 \in \mathcal{V}$. Under the column-vector convention, the forward transition matrix on $[t_{l-1}, t_l]$ is

$$K_l = (1 - b_l)I + b_l e_{\mathsf{m}} \mathbf{1}^\top, \quad l = 1, \ldots, L,$$

so the transition matrix from time 0 to t_l is $\bar{K}_l = K_l K_{l-1} \cdots K_1 = \alpha_l I + (1 - \alpha_l) e_{\mathsf{m}} \mathbf{1}^\top$, for $\alpha_l = \prod_{r=1}^{l}(1 - b_r)$. The conditional one-step time reversal from t_l to t_{l-1} has the closed form

$$K_l^{\leftarrow, x_0} = I + (\alpha_{l-1} - \alpha_l)/(1 - \alpha_l) \cdot (e_{x_0} - e_{\mathsf{m}}) e_{\mathsf{m}}^\top, \qquad l = 1, \ldots, L,$$

with kernel $q_{t_l, t_{l-1}}(y \mid x, x_0) = \mathbb{P}(X_{t_{l-1}} = y \mid X_{t_l} = x, X_0 = x_0) = e_y^\top K_l^{\leftarrow, x_0} e_x$. Equivalently,

$$q_{t_l, t_{l-1}}(y \mid x, x_0) = \begin{cases} (\alpha_{l-1} - \alpha_l)/(1 - \alpha_l), & x = \mathsf{m},\ y = x_0, \\ (1 - \alpha_{l-1})/(1 - \alpha_l), & x = \mathsf{m},\ y = \mathsf{m}, \\ \mathbf{1}\{y = x\}, & x \neq \mathsf{m}, \end{cases}$$

so an already unmasked token remains fixed while a masked token is either kept masked or revealed as its original clean value. Conditioned on the clean sequence, the true one-step reverse kernel factorizes across coordinates, $q_{t_l, t_{l-1}}(\mathbf{x}_{t_{l-1}} \mid \mathbf{x}_{t_l}, \mathbf{x}_0) = \prod_{j=1}^{N} q_{t_l, t_{l-1}}(x_{t_{l-1}}^j \mid x_{t_l}^j, x_0^j)$, which is the analytic posterior that (2.1) approximates. The same derivation covers the reverse kernel used by the standard masked diffusion sampler in Section E.

G Per-output NFE Breakdown

We record the per-output number of function evaluations (NFE), where one NFE counts one sequence-level masked discrete diffusion forward pass, not one vectorized invocation on a batch. This remains a model-only accounting convention: reward evaluations on completed sequences are reported separately and are not included in NFE. Batching can combine many sequence-level evaluations into one physical call without reducing this count; actual latency also depends on hardware, batch size, memory use, and reward-model latency. Let $|\cdot|$ denote set cardinality, s_{split} the number of reverse updates before the best-of-K split, and $[l_{\text{end}}, l^*]$ the guidance window from Algorithm 2. For update index $s \in \{0, \dots, L-1\}$, the corresponding reverse state index is $l = L - s$. Define $\mathcal{G} = \{s : l_{\text{end}} \le L - s \le l^*\}$ as the set of guided updates, with $s \in \{0, \dots, L-1\}$ throughout. Let the active branch multiplicity be $b_s = 1$ for $s < s_{\text{split}}$ and $b_s = K$ for $s \ge s_{\text{split}}$, and set $S = \sum_{s \in \mathcal{G}} b_s$. Under the stated one-head-pass-per-candidate implementation, the per-output model NFE is:

$$\begin{aligned}\text{NFE} = {} & |\{s : s < s_{\text{split}}\}| + K \cdot |\{s : s_{\text{split}} \le s < L\}| \\ & + M \cdot |\{s : s \in \mathcal{G},\, s < s_{\text{split}}\}| + K \cdot M \cdot |\{s : s \in \mathcal{G},\, s_{\text{split}} \le s < L\}|. \end{aligned} \tag{G.1}$$

The first line counts the shared base masked discrete diffusion forward pass on $\bar{\mathbf{x}}_{t_l}$ at every update (one pass per update before the split, K passes per update after). The second line counts the M candidate prediction-head passes for h-estimation inside the guidance window, also with the $K\times$ multiplier after the split. Equivalently, $\text{NFE} = \sum_{s=0}^{L-1} b_s + MS$. Drawing n_h completions per candidate leaves this model NFE unchanged when all completions reuse that candidate's prediction-head probabilities; the executed single-completion sampler uses $n_h = 1$. The corresponding number of scored candidate completions per output is

$$\text{candidate reward evaluations} = n_h M S.$$

If $K > 1$ and a split occurs, terminal ranking requires at most K additional reward evaluations per output when scores of the final sequences are not cached; candidate-completion scores can be reused only when they score those same final sequences. The number of physical batched reward calls is implementation-dependent and is not this sequence-level evaluation count. Setting $K = 1$ or $s_{\text{split}} = L$ disables the split and its terminal ranking, recovering the core D-DOIT branch. These counts describe computational work rather than a wall-clock speed guarantee.

H Detailed Sampler Pseudocode

Algorithm 2 gives the outer sampling loop and Algorithm 3 the per-step guided update.

Algorithm 2 D-DOIT with optional best-of-K refinement. The guided step is defined in Algorithm 3; setting $K{=}1$ or $s_{\text{split}}{=}L$ disables the split. Copies share all tokens revealed before the split; final reward selection changes the output law.

Require: Frozen P_θ, reward R, batch size B, steps L, guidance window $[l_{\text{end}}, l^*]$, split-update index s_{split}, width K, guided-step parameters $(M, \omega, \beta, \tau, \gamma)$

1: Initialize B all-m trajectories at time t_L.
2: **for** $s = 0, 1, \ldots, L-1$ **do**
3: If $s = s_{\text{split}}$ and $K > 1$, duplicate each active trajectory into K copies.
4: Set the reverse state index $l = L - s$.
5: Apply DOOBGUIDEDSTEP to active trajectories when $l_{\text{end}} \le l \le l^*$; otherwise apply the standard transition (2.1).
6: **end for**
7: If $K > 1$ and $s_{\text{split}} < L$, select the highest-reward clean copy in each group by (3.9); otherwise return the generated clean sequences.

Algorithm 3 DOOBGUIDEDSTEP: the practical single-completion update, without proposal-ratio correction.

Require: current state $\bar{\mathbf{x}}_{t_l}$, step index l, frozen model θ, reward R; $(M, \omega, \beta, \tau, \gamma)$

1: Compute α_l and α_{l-1} from the noise schedule {schedule constants}
2: Proposal pass: compute $\mu_\theta^j(\cdot \mid \bar{\mathbf{x}}_{t_l}, t_l)$ for all j
3: Form the τ-tempered proposal head $\mu_{\theta,\tau}^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l) \propto \mu_\theta^j(v \mid \bar{\mathbf{x}}_{t_l}, t_l)^{1/\tau}$
4: Draw $\{\bar{\mathbf{y}}_{t_{l-1}}^{(m)}\}_{m=1}^M$ from (2.1) with μ_θ replaced by $\mu_{\theta,\tau}$ {candidate reverse states}
5: **for** $m = 1, \ldots, M$ **do**
6: Draw $\hat{\mathbf{x}}_0^{(m)} \sim p_{\theta,\gamma}^{\text{head}}(\cdot \mid \bar{\mathbf{y}}_{t_{l-1}}^{(m)})$ via (3.3) {prediction-head completion}
7: $\hat{h}(\bar{\mathbf{y}}_{t_{l-1}}^{(m)}, t_{l-1}) \leftarrow \exp\big(R(\hat{\mathbf{x}}_0^{(m)})/\beta\big)$ {random head-proxy value}
8: **end for**
9: Select m^* by the weights in (3.7); **return** $\bar{\mathbf{x}}_{t_{l-1}} = \bar{\mathbf{y}}_{t_{l-1}}^{(m^*)}$

I Additional Analyses

The main experiments evaluate HepG2 enhancer design, cell-type-specific regulatory DNA design, and protein inverse folding. This appendix collects approximation diagnostics and oracle-transfer analyses; the compute–quality trade-off is discussed in the main text. Matched-time sampling controls are detailed in Sections J.3 and K.5; full diagnostic sweeps appear in Section L.

Concentrated but stochastic selection. At $M = 8$ and $\omega = 8$, mean ESS is 1.143 ± 0.002 and the top-weighted candidate is selected on 95.72% of guided steps. Soft resampling reaches final guidance reward 5.237 ± 0.051, versus 5.154 ± 0.027 for direct argmax in the same diagnostic (Section L.6). Thus selection is near-greedy but not deterministic; low ESS measures local weight concentration, not final-sequence diversity.

Suffix search and nonmonotone progress. With $M = 9$, $K = 4$, and $n_h = 1$ fixed, moving the split from update 105 to 0 increases final guidance reward from 5.272 ± 0.057 to 5.890 ± 0.028, but raises wall time from 326.4 ± 0.5 to 793.6 ± 2.2 seconds per job. Even with 90.38% of bases fixed, independent suffix completions differ at 66.1% of the remaining positions. Late branching therefore searches a diverse suffix conditional on a fixed prefix; it cannot revise pre-split reveals. Separately, held-out activity of argmax-completed partial states is nonmonotone despite higher final activity than frozen sampling (Section L.7). These completion-proxy scores are distinct from final guidance reward R and the batch-median Pred-Activity benchmark. The diagnostics characterize practical search trade-offs, not exact reward-tilted sampling.

J Experimental Details of Regulatory DNA Sequence Design

In this section, we provide the experimental details and additional experimental analysis for the regulatory DNA sequence design task in Section 4.1.

J.1 Dataset and Training Details

Dataset. We conduct experiments on the HepG2 enhancer corpus from Gosai et al. [24]. The corpus contains $\sim$700k 200 bp human enhancer sequences with measured HepG2 activity. We follow the dataset split of Wang et al. [8] so that our evaluation seeds match the DRAKES test partition and our numbers compare against the published DRAKES table.

Reward oracle setup. We use two Enformer-based oracles [64] from Wang et al. [8]. The two oracles use disjoint chromosome splits of the same corpus. The finetuning oracle provides the reward signal $R(\mathbf{x}_0)$ during guided sampling. The held-out evaluation oracle scores Pred-Activity at test time. This matches the DRAKES protocol and makes our numbers comparable to the TDS, SMC, CG, and DRAKES results in Wang et al. [8].

Pre-training setup. D-DOIT operates on a frozen pre-trained masked discrete diffusion model [3]. We do not update its parameters during sampling. We use the public masked discrete diffusion enhancer checkpoint of Wang et al. [8], which the authors pre-train on the $\sim$700k enhancer corpus.

Sampling setup. We use the two operating points defined in Section 3.2: D-DOIT-LBOK-S with a small per-step budget plus a late-stage trajectory split, and D-DOIT-Q with a larger inference-time budget for quality. Table 5 lists the exact hyperparameter values per benchmark. All main DRAKES rows use a single A100 80GB GPU and follow the DRAKES evaluation protocol with 640 sequences per seed. Most rows use 3 random seeds; the re-measured TDS and the adopted D-DOIT-LBOK-S rows use 5 fresh final-test seeds after held-out eval selection. The TDS and D-DOIT-LBOK-S rows are re-measured on an A100-SXM4-80GB GPU; the remaining rows are reported as A100 reruns. Seed identity alone does not guarantee bitwise agreement across hardware or software implementations, so comparisons also require the same checkpoint, sampler, and evaluation protocol. We do not copy numbers from [8], so wall-clock and reward differences against the published table reflect our hardware and seed protocol rather than reporting drift.

J.2 Evaluation Metrics and Baselines

Evaluation metrics. We use four metrics that cover predicted activity, chromatin accessibility, and sequence naturalness:

- **Predicted activity (Pred-Activity):** We use the held-out evaluation reward oracle of [8] to estimate HepG2 enhancer activity for each generated sequence and report the median across the batch following the DRAKES convention. Higher scores indicate more functional sequences.
- **Chromatin accessibility (ATAC-Acc):** We use an independent binary classifier trained on ENCODE HepG2 chromatin accessibility data [65] to test whether the designed sequences fall in accessible chromatin regions. This is a key property of active enhancers. Higher accuracy indicates more active sequences.
- **3-mer pearson correlation (3-mer Corr):** We compute the Pearson correlation between the 3-mer frequency profile of the generated sequences and that of natural HepG2 enhancers from [24]. Higher correlation indicates that the generated sequences match the overall distribution of natural enhancer sequences.
- **JASPAR motif correlation (JASPAR Corr):** We compute the Spearman correlation between the transcription-factor binding motif frequencies of the generated sequences and the natural enhancer reference, scored against the JASPAR motif database [66]. Higher correlation indicates better preservation of regulatory motif structure.

Baselines. We compare D-DOIT against training-free and training-based methods. Training-free methods steer a frozen pre-trained model at inference time:

- **Pre-trained:** the unguided masked discrete diffusion model of [8]; provides the unguided reward and naturalness reference.

Table 5: **Hyperparameters of D-DOIT.** We report two operating points per benchmark: **Q** = quality-oriented and **S** = speed-oriented. Q denotes high-budget quality; S denotes LBOK-S. Selection and test protocols are described in Tables 2 and 3; diagnostic sweeps specify their own configurations.

				DRAKES		Ctrl-DNA	
Sym.	**Name**	**Role**	**Range**	**Q**	**S**	**Q**	**S**
L	Steps	Total discrete reverse steps.	$\{32, 64, 128\}$	128	128	128	128
M	Budget	Candidate next states per guided step; linear head-evaluation cost.	$\{2, \ldots, 64\}$	52	9	52	9
ω	Strength	Exponent on $\hat{h}$ in (3.7); for one completion, log-weight sharpness is ω/β.	$[0.5,\ 8.0]$	3.0	3.0	8.0	3.0
β	Temp.	Scales reward: $\hat{h} = \exp(R/\beta)$ in (3.3). Small β amplifies differences.	$[0.1,\ 2.0]$	0.2	0.2	0.5	0.2
τ	Prop. T	Proposal temperature (3.6) for candidate diversity; changes the reference kernel.	$[1.0,\ 3.0]$	3.0	3.0	3.0	3.0
γ	Comp. T	Completion temperature for prediction-head completion $\hat{\mathbf{x}}_0$ in (3.3); $\gamma<1$ sharpens signal.	$[0.5,\ 2.0]$	1.2	1.2	1.2	1.2
l^*	Guided high	Highest reverse index where guidance is active; guided steps satisfy $l_{\text{end}} \le l \le l^*$.	$\{L/4, \ldots, L\}$	L	L	L	L
l_{end}	Guided low	Lowest reverse index where guidance remains active; guided steps satisfy $l_{\text{end}} \le l \le l^*$.	$\{1, 4, 8\}$	1	1	1	1
s_{split}	Split idx.	Number of reverse updates before each trajectory splits into K copies; 0 applies best-of-K from the start.	$\{0, 0.5L, \ldots, L\}$	0	105	64	105
K	Width	Number of copies per split trajectory (3.9).	$\{1, 2, 4, 8, 20\}$	8	4	20	4
B	Batch	Physical output batch; affects memory use and measured throughput.	—	32	64	256	256
n_h	Completions	Independent head completions per candidate; distinct from sequence length N.	$\{1, 2, 3\}$	1	1	1	1

- **CG** [10]: classifier guidance with discrete gradients; backpropagates through a Gumbel-Softmax relaxation and requires a differentiable reward.
- **SMC** and **TDS** [11]: particle-based guidance with path likelihood ratios. TDS is a strong reported training-free baseline and provides the runtime reference for LBOK-S; its budget is not matched to Q.
- **SVDD-PM** [12]: posterior-mean value decoding, using the same frozen MDLM and oracle protocol, with $M = 416$ for similar runtime to Q.
- **Frozen Best-of-N_{BoN}:** endpoint-only selection over independent unguided samples; $N_{\text{BoN}} = 732$ in the Q wall-clock comparison.

Training-based methods update model parameters:

- **DRAKES** [8]: reward finetunes the pre-trained masked discrete diffusion model with a KL regularizer; we treat it as the strongest naturalness-preserving training-based reference.
- **DRAKES w/o KL**: removes the KL term to expose the over-optimization risk of unregularized reward maximization.
- **CFG** [67]: trains a conditional model from scratch.

J.3 Nearest-neighbor and Endpoint-search Controls

The SVDD-PM comparison uses the same frozen MDLM, guidance Enformer, held-out evaluator, paired seeds 100–102, and 640 final outputs per method and seed. We calibrate its candidate count to $M = 416$ to give a similar wall-clock budget to Q ($L = 128, M = 52, K = 8, s_{\text{split}} = 0$).

Pred-Activity is computed as a median within each seed, then summarized by the mean and sample standard deviation across seeds. The measured times are 10.595 and 11.236 seconds per output for Q and SVDD-PM, respectively. Q has higher means on all four metrics; the clearest separation is ATAC-Acc (91.67% versus 39.84%), while Pred-Activity differs by 0.15.

Table 6: **Matched-time controls.** We test D-DOIT-Q vs SVDD-PM and Frozen Best-of-732 at similar measured wall-clock cost. Bold marks the best mean per column.

Metric	D-DOIT-Q	SVDD-PM	BoN
s/output ↓	10.595	11.236	$\mathbf{10.587}$
Activity ↑	$\mathbf{6.19 \pm 0.05}$	6.04 ± 0.07	4.70 ± 0.03
ATAC (%) ↑	$\mathbf{91.7 \pm 0.9}$	39.8 ± 0.4	43.5 ± 1.3
3-mer ↑	$\mathbf{0.894 \pm 0.009}$	0.688 ± 0.012	0.881 ± 0.008
JASPAR ↑	0.923 ± 0.002	0.917 ± 0.002	$\mathbf{0.957 \pm 0.002}$

Frozen Best-of-N_{BoN} samples independent complete sequences from the same unguided backbone, scores them with the guidance Enformer, and retains the highest-scoring endpoint in each group. A separate held-out Enformer evaluates the retained sequences. Same-A100 calibration gives $N_{\mathrm{BoN}} = 732$, with 10.587 seconds per output compared with Q's 10.595. Table 6 reports this high-budget comparison independently of the lower-budget diagnostics; Q improves activity, accessibility, and 3-mer correlation, while endpoint selection achieves higher JASPAR correlation (0.957 versus 0.923). This control tests whether spending the same time on more complete unguided endpoints can replace local value-guided transitions. Endpoint selection does not reproduce the activity and accessibility gains, although it better preserves this motif statistic. Reward evaluation is included in the measured wall-clock cost.

Separating model and reward budgets. Table 7 additionally compares diagnostic settings with five seeds and 640 outputs per seed. The branched $M = 9$ setting uses $1{,}970$ diffusion evaluations per output, close to Best-of-15's $1{,}920$, but incurs substantially more oracle work. Its $1{,}777$ reward evaluations include $1{,}773$ candidate completions and four terminal scores. Across a 640-output job, batching combines $1{,}137{,}280$ scored sequences into $1{,}290$ vectorized oracle calls. These counts explain why matching only backbone evaluations does not match wall-clock time.

J.4 Guidance-oracle and Evaluator Robustness

We distinguish two controls, both using five sampling seeds and 640 sequences per seed. In Table 8, only the guidance oracle changes among Enformer, CNN, and MLP; sampler settings, seeds, output count, held-out Enformer, and biological metrics remain fixed. All three guidance oracles improve held-out activity and accessibility over the frozen model, but the effect size varies with the oracle. Enformer guidance gives the highest activity and accessibility, whereas MLP guidance gives the highest 3-mer correlation in this diagnostic.

In Table 9, we instead rescore the same Enformer-guided and unguided sequences with held-out Enformer, CNN, and MLP predictors. D-DOIT exceeds the frozen model within each evaluator. Because their output scales differ, cross-column score magnitudes are not directly comparable. These controls assess predictor transfer separately from the selected high-budget Q row and do not substitute for experimental biological assays.

J.5 Additional Experimental Analysis

For the experimental results in Table 2, we provide the additional detailed analysis below.

Section 4.1 reports the headline TDS comparisons. We add two pieces of detail here. First, the D-DOIT-LBOK-S vs. TDS gap on ATAC-Acc is $+25.0$ percentage points ($40.0\% \rightarrow 65.0\%$), while Pred-Activity is matched (4.56 vs. 4.58) with lower reported seed variability; this is descriptive, not a variance-test result. The same operating point also lifts ATAC-Acc from 39.9% to 65.0% over SMC and JASPAR Corr from 0.756 to 0.948 over SMC. Second, high-budget D-DOIT-Q over TDS gives $+1.61$ in Pred-Activity, $+51.7$ percentage points in ATAC-Acc, $+0.077$ in 3-mer Corr, and $+0.124$ in JASPAR Corr.

Table 7: **Backbone work, oracle work, and measured time on DRAKES.** Each setting uses five sampling seeds and 640 outputs per seed. Pred-Activity is held-out Enformer activity; entries report means and sample standard deviations across seeds. The final column measures the full 640-output job and must not be extrapolated to other physical batch sizes. The unbranched and branched rows are diagnostic settings, not substitutions for the separately selected main-table LBOK-S or Q results.

Configuration	Pred-Activity ↑	Model NFE per output	Reward evaluations per output	s/640 outputs
Frozen Best-of-15	2.219 ± 0.017	1,920	15	97.8 ± 1.5
$M = 9, \tau = 1, K = 1$	3.115 ± 0.021	1,280	1,152	227.0 ± 1.9
$M = 9, \tau = 3, K = 1$	3.670 ± 0.072	1,280	1,152	227.8 ± 2.6
$M = 9, \tau = 3, K = 4, s_{\text{split}} = 105$	4.491 ± 0.045	1,970	1,777	327.8 ± 4.3

Table 8: **Changing the guidance oracle.** Only the guidance reward R changes among the D-DOIT rows; the sampler, seeds, output count, held-out Enformer evaluator, and other metrics are fixed. Entries are mean $\pm$ sample standard deviation over five seeds, with 640 sequences per seed. Pred-Activity uses the held-out Enformer. These results are separate from the high-budget Q evaluation.

Generation method	Pred-Activity ↑	ATAC-Acc (%) ↑	3-mer Corr ↑	JASPAR Corr ↑
Enformer-guided D-DOIT	4.491 ± 0.045	68.25 ± 1.86	0.8630 ± 0.0160	0.9468 ± 0.0025
CNN-guided D-DOIT	2.203 ± 0.054	51.91 ± 1.19	0.8738 ± 0.0013	0.9342 ± 0.0021
MLP-guided D-DOIT	1.544 ± 0.037	28.78 ± 1.29	0.9489 ± 0.0035	0.9347 ± 0.0031
Frozen pre-trained model	0.371 ± 0.018	1.66 ± 0.59	-0.0584 ± 0.0149	0.2723 ± 0.0220

Compared with the training-based DRAKES, D-DOIT-Q achieves higher Pred-Activity (6.19 vs. 5.61), higher 3-mer Corr (0.894 vs. 0.887), and higher JASPAR Corr (0.923 vs. 0.911) without any backbone update. DRAKES remains higher on ATAC-Acc (92.5% vs. 91.7%), so we do not claim an all-metric win. The unregularized variant DRAKES w/o KL exposes the cost of training-based reward maximization: it gains $+0.83$ in Pred-Activity over DRAKES but loses 0.580 in 3-mer Corr ($0.887 \to 0.307$) and 0.354 in JASPAR Corr ($0.911 \to 0.557$). D-DOIT-Q stays near the DRAKES high-naturalness regime. The frozen model supplies candidate proposals without retraining. The exact reference transform reweights its induced distribution; the practical sampler's temperatures and selection operations change that reference law. Figure 2 plots utility and naturalness for the original benchmark comparison set. The separately time-calibrated SVDD and endpoint controls are reported in Table 6, rather than pooled into a claim of an exhaustive compute–quality frontier. D-DOIT-LBOK-S anchors the practical moderate-budget side of this frontier. D-DOIT-Q occupies the high-utility, high-naturalness corner when inference-time budget is deliberately relaxed, with ATAC-Acc as the remaining gap to DRAKES.

J.6 Held-out Final-test Check for D-DOIT-LBOK-S

We select the DRAKES LBOK-S configuration on eval seeds $42, 43, 44$ and report it once on fresh final-test seeds $105, 106, 107, 108, 109$. The fixed final configuration uses M=9, K=4, s_{split}=105, ω=3.0, β=0.2, τ=3.0, and γ=1.2. It exceeds the TDS thresholds on ATAC-Acc, 3-mer Corr, and JASPAR Corr, and has a similar mean Pred-Activity to TDS; no statistical equivalence test is implied.

J.7 Held-out Selection for High-budget D-DOIT-Q

To avoid tuning directly on the final reported seeds, we separate the DRAKES high-budget update into three seed groups. Development seeds $0, 1, 2$ identify promising sampler families only. Held-out eval seeds $42, 43, 44$ select the final high-budget D-DOIT-Q configuration (M=52, K=8, s_{split}=0, ω=3.0, β=0.2, τ=3.0, γ=1.2). Final-test seeds $100, 101, 102$ are then evaluated once and reported as the held-out final numbers in Table 2.

Table 9: **Transfer across held-out evaluators.** The same Enformer-guided D-DOIT and frozen-model sequences are rescored by three held-out predictors. Entries are mean ± sample standard deviation over five seeds, with 640 sequences per seed. Enformer scores are Pred-Activity; CNN and MLP scores have their own scales. Compare methods within each column, not absolute values across columns.

Generation method	Pred-Activity ↑ (held-out Enformer)	Held-out CNN ↑	Held-out MLP ↑
Enformer-guided D-DOIT	4.491 ± 0.045	0.867 ± 0.014	0.634 ± 0.015
Frozen pre-trained model	0.371 ± 0.018	0.484 ± 0.014	0.509 ± 0.014

Table 10: **Held-out final-test check for D-DOIT-LBOK-S.** Targets are the TDS row in Table 2. The labels compare reported means with the reference values; they are not significance or statistical-equivalence decisions.

Metric	TDS reference	D-DOIT-LBOK-S final test	Mean comparison
Pred-Activity	4.58	4.560(0.040)	similar mean
ATAC-Acc	40.0%	65.03(1.43)%	pass
3-mer Corr	0.817	0.8511(0.0165)	pass
JASPAR Corr	0.798	0.9477(0.0016)	pass

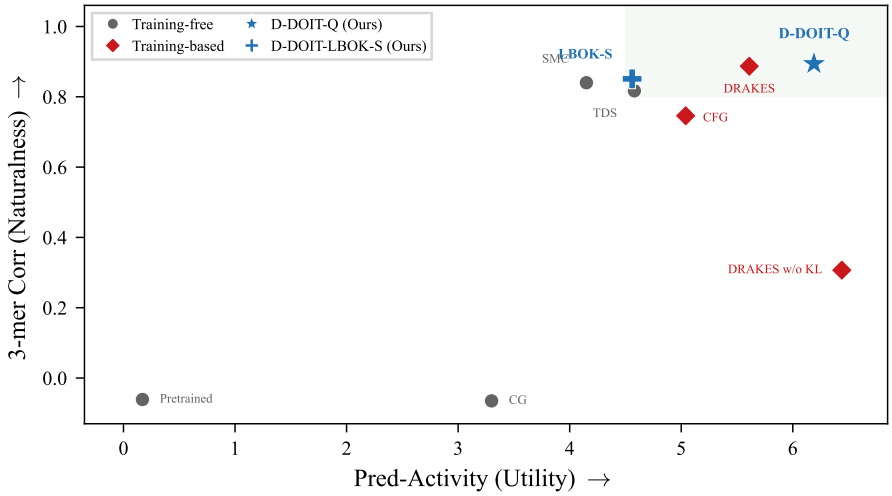


Figure 2: **Utility-naturalness trade-off.** We plot Pred-Activity (median) against 3-mer Corr; one point per method. Green shading marks the high-utility, high-naturalness region. This view covers the original benchmark rows; the separately time-calibrated SVDD and endpoint comparisons are in Table 6. ATAC-Acc is reported separately in Tables 2 and 11.

Table 11: **Held-out final-test check for D-DOIT-Q.** Targets are the DRAKES row in Table 2. The final-test result passes Pred-Activity, 3-mer Corr, and JASPAR Corr, but misses the ATAC-Acc target on the held-out evaluation.

Metric	DRAKES reference	D-DOIT-Q final test	Mean comparison
Pred-Activity	5.61	6.19(0.05)	pass
ATAC-Acc	92.5%	91.7(0.9)%	miss
3-mer Corr	0.887	0.894(0.009)	pass
JASPAR Corr	0.911	0.923(0.002)	pass

K Experimental Details of Cell-type-specific Enhancer Design

In this section, we provide the experimental details and additional experimental analysis for the cell-type-specific enhancer design task in Section 4.2.

K.1 Dataset and Training Details

Dataset. We use the 200 bp human enhancer corpus of Chen et al. [25], covering three cell lines: HepG2, K562, and SK-N-SH. Data preprocessing follows the regLM protocol of Lal et al. [26], as in Chen et al. [25]. We use the Ctrl-DNA enhancer task only for its data, cell-type-specific reward oracles, and evaluation metrics. The training procedure and backbone architecture come from our own setup, which we describe in the next paragraphs.

Reward oracle setup. We use the cell-type-specific Enformer-based oracles [64] from Chen et al. [25], which the authors train on the same corpus. Unlike constrained-RL baselines with explicit thresholds and policy retraining, D-DOIT is an inference-time method and accepts a single black-box scalar reward. For target cell line c and off-target cell lines $\mathcal{C}_{-c}$, we set

$$R(\mathbf{x}_0) = R_c(\mathbf{x}_0) - \lambda \cdot \max_{c' \in \mathcal{C}_{-c}} R_{c'}(\mathbf{x}_0), \tag{K.1}$$

where each R_c is the cell-type-specific Enformer oracle of Chen et al. [25]. The $\max$ over off-targets suppresses the worst case during sampling. The evaluation ΔR averages over off-targets to follow the Ctrl-DNA convention. We sweep λ on held-out eval seeds disjoint from the final-test seeds. The final high-budget row uses the task-default penalties selected for the Q operating point, and the final low-budget LBOK-S row uses $\lambda = 2.0$ selected by the low-budget eval sweep.

Pre-training setup. We train a new masked discrete diffusion model [3] on the Ctrl-DNA enhancer corpus. The HyenaDNA backbone used by Chen et al. [25] is autoregressive and does not fit our discrete h-transform. D-DOIT operates on this frozen model without updating its parameters during sampling. The same checkpoint serves all three cell lines and both operating points.

Sampling setup. We use the two operating points from Section 3.2. Table 5 lists the exact hyperparameter values per benchmark. For the D-DOIT Ctrl-DNA rows, we choose fixed configs on eval seeds $42, 43, 44$ and report final-test seeds $100, 101, 102, 103, 104$. We use 5 random seeds with batch size 256 per seed and run a separate set for each of the three target cell lines. Pre-trained, CG, SMC, and TDS are the rerun training-free comparison rows. Other baselines come from the reported results in Chen et al. [25] Table 1 under the 5-seed, $\delta = 0.5$ setting.

K.2 Evaluation Metrics and Baselines

Evaluation metrics. We use three metrics that cover cell-type specificity, motif structure, and set-level diversity:

- **Cell-specific reward difference (ΔR):** We compute the difference between the target-cell oracle score and the average of the off-target oracle scores following the Ctrl-DNA convention. Higher scores indicate stronger cell-type specificity.
- **Motif correlation (Motif Corr):** We compute the Pearson correlation between the transcription-factor binding-site frequencies of the generated sequences and the cell-type-specific natural enhancer set from [25] (top half by target activity, bottom half by off-target activity). Higher correlation indicates that the generated sequences match the regulatory grammar of natural cell-specific enhancers.
- **Set-level diversity (Diversity):** We compute the Shannon entropy across the generated sequence set with the official Ctrl-DNA evaluation script [25]. Higher entropy indicates that the sampler does not collapse onto a small set of high-reward sequences.

ΔR and Motif Corr appear in Table 3. Diversity appears in Figure 3. We omit Median because [25] mentions it in text but does not report numerical values.

Baselines. We compare D-DOIT against the baselines from Chen et al. [25] and the same training-free guidance methods as in Section J.2:

Table 12: **Ctrl-DNA training-free baseline tuning screen.** We run a bounded seed-42 screen over 13 CG, SMC, and TDS hyperparameter settings on all three Ctrl-DNA cell tasks. The screen documents that these training-free baselines remain weak after tuning: CG preserves motifs without improving reward/specificity, SMC gains little reward while degrading motif/diversity, and TDS stays near zero reward movement. The selected configs are provenance for possible follow-up evaluation, not final-test rows.

Method	Seed-42 screen	Selected config	Screen outcome	Takeaway
CG	guidance scale $\{20, 50, 100, 200\}$	CG_G20: guidance scale 20	$\Delta R \in [-0.002, 0.000]$; specificity ≤ -0.007; motif corr 0.903–0.916	All screened CG settings preserve motifs but do not improve reward or specificity.
SMC	beta values: 0.3, 0.5, 0.75, 1.0, 1.5	SMC_B05: beta 0.5	Best $\Delta R = 0.011$ at beta 1.5, with motif corr 0.388 and diversity 0.928; selected beta 0.5 has $\Delta R = 0.006$.	Small reward gains come with much weaker motif/diversity, so SMC is not competitive.
TDS	g/beta pairs: $2/0.75$, $5/0.75$, $5/1.0$, $10/1.0$	TDS_G5_B10: $g = 5$, beta 1.0	$\Delta R \in [-0.002, 0.001]$; specificity < 0 for all settings; best motif corr 0.510.	TDS remains weak after this bounded tuning pass.

- **Pre-trained**, **CG** [10], **SMC**, **TDS** [11]: training-free inference-time methods on the same frozen masked discrete diffusion model. These share the frozen backbone with D-DOIT. Frozen Best-of-708 additionally provides the same-A100 endpoint-search control at Q's measured time budget.
- **regLM** [26]: supervised generative baseline that the authors train on the cell-type-specific enhancer corpus.
- **PPO, TACO, PPO-Lagrangian** [25]: RL-based methods that retrain the generator against the specificity reward.
- **AdaLead, PEX, BO, CMA-ES** [25]: directed-evolution methods that search the sequence space directly without any generative backbone.

K.3 Training-free Baseline Tuning Screen

To check that the Ctrl-DNA training-free baselines are not weak only because of arbitrary default hyperparameters, we run a bounded seed-42 screen for CG, SMC, and TDS on HepG2, K562, and SK-N-SH. Table 12 summarizes all 13 screened configurations at the family level and names the selected follow-up configuration for each baseline family. Across this screen, CG remains a high-motif, near-zero-reward baseline; SMC obtains only small reward movement and degrades motif/diversity; and TDS remains weak on the composite specificity reward. Because this is a one-seed screen rather than a final-test protocol, we use it as baseline-tuning provenance instead of replacing the main-table baseline rows reported above.

K.4 Additional Experimental Analysis

For the experimental results in Table 3, we provide the additional detailed analysis below.

Table 3 records positive mean specificity for the two D-DOIT configurations relative to the originally reported CG, SMC, and TDS rows. The time-calibrated endpoint control additionally attains positive specificity, with mean $\Delta R = 0.108$. Compared with TDS, D-DOIT-LBOK-S improves ΔR from 0.002 to 0.033 and improves Motif Corr from 0.573 to 0.648, at about $0.94\times$ the rerun TDS wall-clock in the Ctrl-DNA timing logs. CG fails to move the specificity reward at all ($\Delta R \approx 0$, indistinguishable from Pre-trained), and SMC sits at $\Delta R = 0.000$ with Motif Corr 0.587. Q reaches $\Delta R = 0.119 \pm 0.063$ and Motif Corr 0.498 ± 0.070. Its 5.43 seconds per output is close to Frozen Best-of-708's 5.51 on the same A100. The small mean advantage over endpoint selection is not claimed statistically significant. Table 13 provides the per-cell breakdown for the original guidance comparisons.

Compared with the remaining training-based baselines, D-DOIT-Q stays behind regLM on ΔR (0.119 vs. 0.16). Because regLM uses a different generative backbone and training procedure, this gap does not isolate the effect of inference-time adaptation. On Motif Corr, D-DOIT-Q at 0.498 exceeds every remaining training-based and evolutionary baseline (PPO-Lagrangian 0.42,

AdaLead 0.36). These means describe motif preservation at the reported specificity levels; they do not identify which mechanism caused the difference. Pre-trained has high Motif Corr (0.937) at $\Delta R \approx 0$, providing an unguided naturalness reference rather than a mathematical ceiling. LBOK-S has higher Motif Corr (0.648) than Q and Frozen Best-of-708, while achieving lower specificity. D-DOIT-Q (0.498) trades motif fidelity for higher ΔR, so we use it as the high-specificity operating point rather than a motif-preserving one.

Figure 3 compares diversity and specificity within the original guidance baseline set; the endpoint control's reward and motif scores are compared separately in Table 3. SMC and TDS gain little ΔR and collapse Shannon entropy to ≈ 1.5. D-DOIT-Q occupies the high-ΔR, high-diversity region across all three cell lines. Table 13 reports the per-cell breakdown of ΔR, Motif Corr, and Diversity.

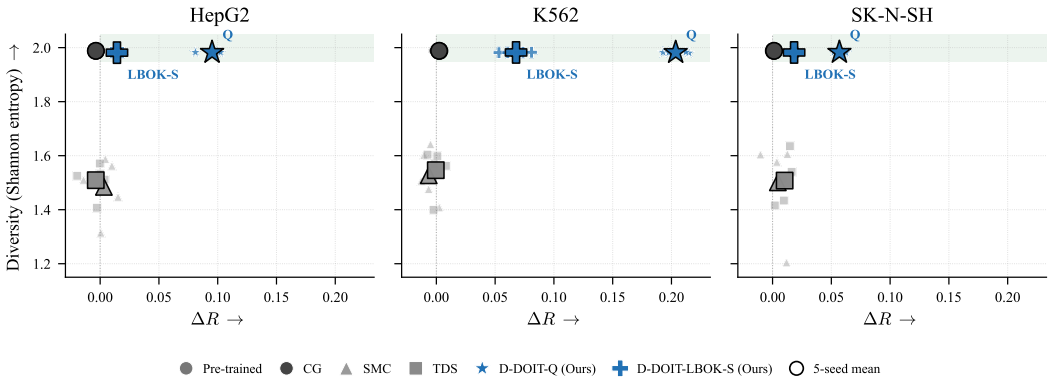


Figure 3: **Specificity vs. diversity on Ctrl-DNA.** Each panel is one target cell line. Small markers are single seeds, and large black-edged markers are the corresponding 5-seed means. The x-axis is cell-specific reward gain ΔR; the y-axis is set-level Shannon entropy from the Ctrl-DNA evaluation script [25]. The green region marks positive ΔR with high diversity. D-DOIT moves into this region, whereas Pre-trained/CG remain near zero reward movement and SMC/TDS lose diversity.

The headline averages in Table 3 are the cross-cell means of these rows. Both D-DOIT operating points achieve their largest ΔR on K562, consistent with K562 carrying the largest reward headroom in the Ctrl-DNA setup, with smaller positive gains on HepG2 and SK-N-SH. D-DOIT-LBOK-S wins on Motif Corr for every cell line among training-free methods that move the reward. Both D-DOIT operating points keep Diversity high (≈ 1.98) on all three cells. SMC and TDS drop to ≈ 1.5.

The final Ctrl-DNA Q operating point uses $M = 52$, $K = 20$, and $s_{\text{split}} = 64$, whereas DRAKES Q uses $M = 52$, $K = 8$, and $s_{\text{split}} = 0$. A preceding Ctrl-DNA eval sweep also considered $M = 8, K = 20, s_{\text{split}} = 64$, which obtained config-level $\Delta R = 0.117$, and a $M = 32, K = 8, s_{\text{split}} = 0$ setting with $\Delta R = 0.105$ at roughly three times the runtime. These exploratory settings differ from the final Q comparison and vary several parameters together. They illustrate alternative budget allocations, not an isolated causal effect of branching. The endpoint control shows that final selection alone already obtains much of the mean specificity gain under the calibrated budget.

K.5 Wall-clock-matched Endpoint Control

Frozen Best-of-708 generates complete unguided sequences from the frozen backbone and retains the best reward-scored endpoint in each group. We calibrate its group size on the same A100 used for Q ($L = 128, M = 52, K = 20, s_{\text{split}} = 64$). At 5.51 versus 5.43 seconds per output, the methods give specificity 0.108 ± 0.0281 versus 0.119 ± 0.063, and Motif Corr 0.490 ± 0.165 versus 0.498 ± 0.070. The means favor Q slightly, but the differences are modest and are not interpreted as statistically significant. Training-based rows remain cross-backbone references, separate from this inference-time comparison.

Table 13: **Per-cell-type Ctrl-DNA results.** We break down ΔR, Motif Corr, and Diversity by target cell line. Parentheses denote standard deviation across 5 random seeds. **Bold** = best, underline = second best, per column within each cell line.

Cell line	Method	$\Delta R\uparrow$	Motif Corr $\uparrow$	Diversity $\uparrow$
HepG2	Pre-trained	$-0.004(0.002)$	$0.937(0.009)$	$\mathbf{1.988(0.001)}$
	CG	$-0.004(0.002)$	$\mathbf{0.938(0.009)}$	$\mathbf{1.988(0.001)}$
	SMC	$+0.003(0.011)$	$0.582(0.094)$	$1.484(0.109)$
	TDS	$-0.004(0.009)$	$0.606(0.087)$	$1.509(0.061)$
	D-DOIT-LBOK-S (Ours)	$+0.014(0.003)$	$0.667(0.032)$	$1.981(0.002)$
	D-DOIT-Q (Ours)	$\mathbf{+0.095(0.008)}$	$0.558(0.048)$	$1.982(0.001)$
K562	Pre-trained	$+0.003(0.003)$	$0.937(0.009)$	$\mathbf{1.988(0.001)}$
	CG	$+0.002(0.003)$	$\mathbf{0.938(0.009)}$	$\mathbf{1.988(0.001)}$
	SMC	$-0.006(0.006)$	$0.613(0.133)$	$1.527(0.095)$
	TDS	$-0.000(0.006)$	$0.538(0.072)$	$1.546(0.084)$
	D-DOIT-LBOK-S (Ours)	$+0.068(0.010)$	$0.620(0.031)$	$1.982(0.001)$
	D-DOIT-Q (Ours)	$\mathbf{+0.204(0.009)}$	$0.421(0.037)$	$1.982(0.001)$
SK-N-SH	Pre-trained	$+0.001(0.001)$	$0.937(0.009)$	$\mathbf{1.988(0.001)}$
	CG	$+0.001(0.001)$	$\mathbf{0.938(0.010)}$	$\mathbf{1.988(0.001)}$
	SMC	$+0.004(0.009)$	$0.566(0.133)$	$1.501(0.170)$
	TDS	$+0.010(0.006)$	$0.573(0.075)$	$1.508(0.088)$
	D-DOIT-LBOK-S (Ours)	$+0.018(0.002)$	$0.658(0.054)$	$1.982(0.001)$
	D-DOIT-Q (Ours)	$\mathbf{+0.057(0.003)}$	$0.517(0.035)$	$1.983(0.001)$

Table 14: **Evolutionary-search baselines on Ctrl-DNA.** We report ΔR and Motif Corr averaged across the three target cell lines for the four directed-evolution baselines reproduced from [25]. The training-free and training-based comparisons appear in Table 3.

Method	$\Delta R\uparrow$	Motif Corr $\uparrow$
AdaLead	$0.05(0.05)$	$0.36(0.08)$
PEX	$0.09(0.02)$	$0.23(0.04)$
BO	$0.01(0.02)$	$0.11(0.11)$
CMA-ES	$0.01(0.01)$	$0.08(0.05)$

L Ablation Study

We conduct ablation studies on the DRAKES HepG2 enhancer task to understand the behavior of D-DOIT. We examine the guidance window l^*, guidance strength ω and effective sample size (ESS), proposal and completion temperatures τ, γ, and candidate budget M. We also study completion count, direct-versus-rollout value estimates, soft-versus-argmax selection, split position, and reward trajectories (Tables 16 to 21). The window, strength, temperature, and candidate-budget sweeps use a single held-out seed unless noted otherwise; the additional diagnostics state their sampling protocols separately. Final guidance reward R is scored by the sampling oracle and is distinct from held-out Pred-Activity. Completion-proxy trajectories use the held-out evaluator but average scores of completed partial states rather than the benchmark's within-seed batch median. We re-run the best γ-sweep configuration across the 3 main evaluation seeds and report it in Table 15. These studies show how D-DOIT balances reward optimization, sequence naturalness, and computational cost.

The candidate-budget sweep shows that increasing M mainly improves functional utility, while naturalness metrics remain comparatively stable. The table surfaces three consequential design choices: the guidance window l^* exhibits a phase transition for naturalness; the completion temperature γ controls prediction-head completion quality; and the candidate budget M controls utility.

Table 15: **Headline ablation findings.** Each row varies one hyperparameter around the DRAKES ablation reference setting. We report sweep-margin endpoints from a single-seed sweep; full sweeps appear in Section L.

Axis	Setting	3-mer Corr	Pred-Activity / ATAC (%)
l^*	$64 \to 128$	$-0.06 \to 0.894$	$3.61 \to 4.77$ / $33.6 \to 72.7$
γ	$1.2 \to 0.6$	$0.846 \to 0.905$	$4.53 \to 4.72$ / $72.7 \to 66.4$
M	$8 \to 32$	$+9\%$	$+31\%$ / $+29$ pp

Runtime analysis. Section 4.4 reports the runtime comparison and its measurement conditions. The runtime pattern follows the NFE accounting in Section G: larger M and earlier best-of-K splitting increase candidate-state evaluations and oracle-scored completions. Increasing n_h also increases completion scoring, as quantified in Table 16. LBOK-S is the lower-budget operating point, whereas Q deliberately trades greater recurring inference cost for quality; neither setting implies cost dominance over training-based methods.

L.1 Effect of Guidance Window l^*

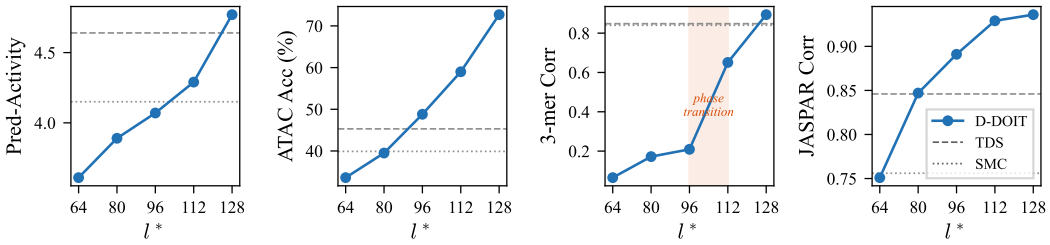


Figure 4: **Guidance-window sweep.** Expanding the guided window improves utility and sharply recovers naturalness between $l^* = 96$ and $l^* = 112$.

We study the effect of the guidance window l^*, the highest reverse-time index where Doob's h-transform is active in the discrete sampler. A smaller l^* saves $(L - l^*) \cdot M$ neural-function evaluations per sequence but omits guidance at more heavily masked states, where limited revealed context may affect the prediction-head proxy. Figure 4 reports the four DRAKES metrics as we sweep $l^* \in \{10, 20, \ldots, 128\}$.

The results show a phase-transition-like behavior in naturalness. All configurations with $l^* \leq 50$ produce negative 3-mer Pearson correlations. A sharp jump occurs between l^*=96 (3-mer = 0.21)

and l^*=112 (0.65). At l^*=128, 3-mer reaches 0.894, a 14× improvement over l^*=64. Utility metrics improve across the same range without a sharp transition. In this sweep, broad guidance windows are associated with better global sequence-composition statistics; this observation does not isolate a causal mechanism or establish a requirement for other rewards. Utility metrics, in contrast, recover from a narrower window. We use l^*=128 as the default setting.

L.2 Effect of Guidance Strength ω and ESS Collapse

We study the effect of the guidance strength ω in (3.7). A larger ω sharpens the transformed transition toward high-reward candidates. A smaller ω preserves more of the pre-trained kernel.

Figure 5 reports the per-step Effective Sample Size (ESS) of the candidate weights and the fraction of steps where ESS collapses below $M/4$. The results show that ESS collapses to near 1 at $\omega \geq 4$. 95% of guided steps have ESS $< M/4$ at ω=4, and 100% at $\omega \geq 16$. A direct mechanism for concentration is reward-range amplification: ω=4 amplifies a candidate reward range of $\approx$ 3 nats to $\geq$ 12 nats and saturates the softmax. High-quality results remain achievable despite this collapse when proposal-level diversity (via τ) still places a strong candidate in the pool. This motivates joint tuning of ω and τ, but does not imply that larger τ recovers importance-weight accuracy. The concentration depends on the reward distribution and ω/β, so these diagnostic thresholds cannot be transferred unchanged to other configurations.

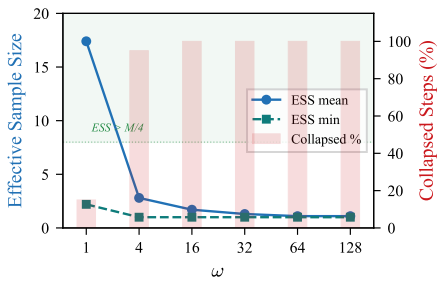


Figure 5: **ESS diagnostic.** ESS and collapse fraction across ω; weighted selection collapses to near-greedy at $\omega \geq 4$.

L.3 Effect of Proposal and Completion Temperatures τ, γ

We then study the joint effect of the proposal temperature τ in (3.6) and the completion temperature γ in (3.3). A larger τ flattens the proposal and produces more diverse candidates. A smaller γ sharpens the prediction-head completion toward the model mode. Figure 6 reports the four DRAKES metrics across the joint sweep at M=32.

The results show a clear separation of roles. Functional activity peaks at higher proposal diversity. Marginal analysis shows Pred-Activity improves from 3.89 at τ=2.5 to 4.07 at τ=5.0, and ATAC-Acc from 53.7% to 69.4%. Naturalness peaks at sharper prediction-head completions. Reducing γ from 1.2 to 0.6 improves averaged 3-mer Corr from 0.846 to 0.905. At the strongest γ-sweep configuration (M=32, τ=3.0, γ=0.6), 3-mer Corr reaches 0.949 $\pm$ 0.006 across 3 seeds, above DRAKES at 0.887. This sweep identifies the naturalness-preserving region. The final held-out DRAKES Q row in Table 2 uses a larger eval-selected inference budget and trades some naturalness for much higher Pred-Activity and ATAC-Acc while remaining above the DRAKES 3-mer and JASPAR targets. Naturalness metrics saturate around τ=3.5–4.0, suggesting a trade-off between naturalness and accessibility at extreme proposal temperatures. We use τ=3.0 and γ=0.6 as the ablation default for naturalness-preserving sweeps.

L.4 Effect of Candidate Budget M

We study the effect of the Monte Carlo candidate budget M. The candidate count controls per-step work and coverage of possible next states. This sweep measures quality changes, not kernel error or convergence to the original tilted distribution. Figure 7 reports the four DRAKES metrics across $M \in \{8, 16, 32\}$.

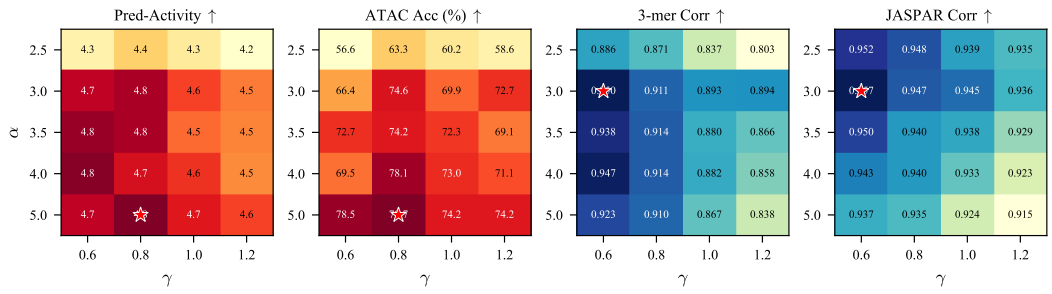


Figure 6: **Interaction of proposal and completion temperatures.** We sweep the proposal temperature τ and the completion temperature γ at $M = 32$. Naturalness peaks at sharper prediction-head completions (smaller γ), while functional activity peaks at higher proposal diversity (larger τ). The red star marks the ablation reference configuration used before the final held-out Q budget selection.

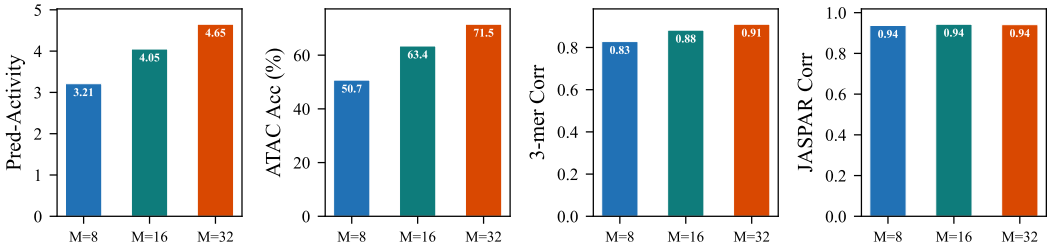


Figure 7: **Candidate-budget sweep.** We vary the Monte Carlo candidate budget M on the DRAKES HepG2 task and report the same four metrics. Increasing M gives direct gains in functional utility, while naturalness changes much less across the sweep.

The results show that fewer candidates reduce functional utility but leave sequence naturalness almost unchanged. Reducing M from 32 to 8 drops Pred-Activity by 31% and ATAC-Acc by 29%. 3-mer Corr drops by 9% and JASPAR Corr by less than 1%. The temperatures and the guidance window drive naturalness. A larger candidate count gives more direct help to utility. M=32 provides a strong per-step default within the ablation sweep. The final held-out DRAKES Q row uses a larger budget, M=52 with K=8, for the no-budget-limit quality comparison in Table 2.

L.5 Completion Count and Matched-state Proxy Diagnostics

Completion count n_h and candidate count M address different aspects of sampling: M controls candidate-state coverage, whereas n_h controls the number of prediction-head completions averaged for each candidate in (3.5). We reserve N for sequence length. Table 16 reports the full sampler with five seeds and 3,200 sequences per setting. Increasing n_h from 1 to 2 raises final guidance reward by 0.492, at 41% additional wall time; $n_h = 3$ raises reward by 0.706, at 81% additional wall time. We retain $n_h = 1$ as the practical compute–quality default. More completions reduce sampling variance for the head expectation, but do not remove its factorization or temperature mismatch with sequential rollouts. The completion-count sweep therefore measures a practical quality–cost trade-off while retaining the same factorized head approximation.

To examine the completion law, we save 640 matched partial states at each of three stages from 128 DRAKES HepG2 source trajectories per seed over five seeds. From each state, we independently generate 32 direct prediction-head completions and 32 sequential reverse-process continuations. For each route, we compute the h-value corresponding to each completion's reward and average these values over the 32 completions. Table 17 summarizes the pooled distributions of these per-state averages. The fixed-base fraction is the percentage of nucleotides already revealed in the partial state. W1 is Wasserstein-1 distance; KS is the maximum absolute difference between the two empirical cumulative distribution functions, not a significance-test result.

Table 16: **Completion-count ablation on HepG2**. We use five seeds and 3,200 sequences per setting. The score is final guidance reward R, not held-out Pred-Activity. Runtime increases are relative to $n_h = 1$.

Completions n_h	Final guidance reward R	Wall-time increase
1	5.272 ± 0.057	baseline
2	5.764 ± 0.033	41%
3	5.978 ± 0.026	81%

Table 17: **Matched-state completion diagnostic.** At each stage, 640 partial states come from 128 source trajectories per seed over five seeds. Each state is independently completed 32 times by the direct prediction head and 32 times by sequential continuation. Moments and distances summarize the pooled distributions of per-state average h-values; parentheses denote variance, not uncertainty.

Fixed bases	Direct: mean (variance)	Rollout: mean (variance)	W1	KS
0.78%	3.494 (0.974)	3.480 (1.097)	0.060	0.045
33.39%	3.536 (1.241)	3.611 (1.109)	0.101	0.050
66.34%	3.648 (1.685)	3.772 (1.620)	0.124	0.058

The pooled means are similar, while W1 increases from 0.060 at 0.78% fixed bases to 0.124 at 66.34%. KS remains between 0.045 and 0.058. Thus the direct head gives a useful low-cost marginal approximation in this diagnostic, with residual distributional shift that grows across the sampled stages. These pooled summaries do not imply equality of the conditional completion laws or exact sampling from the reward-tilted target.

L.6 Soft Selection versus Argmax

Table 18 compares soft selection in (3.7) with direct argmax using $M = 8$, $\tau = 3$, $K = 4$, and $\beta = 0.2$, over five seeds and 3,200 sequences per setting. This diagnostic is separate from the high-budget Q operating point. As ω rises from 1 to 8, mean ESS falls from 1.877 to 1.143 and top-candidate selection rises from 77.53% to 95.72%. At $\omega = 1$, soft selection has lower final guidance reward than argmax (4.894 ± 0.055 versus 5.154 ± 0.027); at $\omega = 8$, it has higher reward (5.237 ± 0.051). The behavior is therefore close to stochastic best-of-M selection, rather than broad resampling, but not identical to argmax. The remaining 4.28% non-top selections at $\omega = 8$ can change subsequent states and candidate pools. Low ESS describes local weight concentration, not final-sequence diversity or target-distribution accuracy.

Table 18: **Soft selection versus direct argmax on HepG2.** We use $M = 8$, $\tau = 3$, $K = 4$, and $\beta = 0.2$, with five seeds and 3,200 sequences per setting. ESS and top-candidate frequency describe guided-step selection; rewards are final guidance reward R. Direct argmax does not depend on ω.

ω	Mean ESS	Largest normalized weight	Top-candidate selection rate	Soft reward R	Argmax reward R
1	1.877 ± 0.020	0.775	77.53%	4.894 ± 0.055	5.154 ± 0.027
4	1.237 ± 0.004	0.927	92.69%	5.183 ± 0.038	5.154 ± 0.027
8	1.143 ± 0.002	0.957	95.72%	5.237 ± 0.051	5.154 ± 0.027

L.7 Split Time and Reward Trajectories

Earlier branching improves reward at greater cost. Table 19 varies only s_{split}, keeping $M = 9$, $K = 4$, and $n_h = 1$ fixed. Splitting at update 0 gives final guidance reward 5.890 ± 0.028 and a best-of-4 gain of 1.605 ± 0.019; splitting at update 105 gives 5.272 ± 0.057 and 0.959 ± 0.020, respectively. The earlier split costs 793.6 ± 2.2 seconds per job, compared with 326.4 ± 0.5 seconds for the late split. Late branching retains a useful selection gain while sharing more of the trajectory computation, but it cannot revise tokens already fixed before the split.

Table 19: **Controlled split-position ablation with $M = 9$, $K = 4$, and $n_h = 1$ fixed.** Only s_{split} varies. Both final reward and best-of-4 selection gain use the guidance reward R, not held-out Pred-Activity.

Split update s_{split}	Final guidance reward R	Best-of-4 gain	Wall time/job (s)
0	5.890 ± 0.028	1.605 ± 0.019	793.6 ± 2.2
32	5.855 ± 0.035	1.545 ± 0.030	653.8 ± 1.5
64	5.729 ± 0.029	1.435 ± 0.019	510.4 ± 1.9
105	5.272 ± 0.057	0.959 ± 0.020	326.4 ± 0.5

Diversity remains in the unresolved suffix. For saved partial states from the frozen reverse process, we sample 32 independent suffix completions and compute pairwise Hamming distance only on positions that were still masked (Table 20). The distance is 0.710 ± 0.010 with 25.43% of bases fixed, and remains 0.661 ± 0.011 with 90.38% fixed. These results demonstrate substantial suffix diversity conditional on a shared prefix, not determinism of early reveals. Late best-of-K exploits this remaining search space; the complete D-DOIT–TDS comparison does not isolate branching's contribution.

Table 20: **Suffix diversity under the frozen reverse process.** For each saved partial sequence, 32 independent suffix completions are sampled. Hamming distance is the mean pairwise fraction of differing bases restricted to positions that were still masked in the shared partial state.

Fixed positions	Still-masked positions	Remaining-position Hamming distance
25.43%	74.57%	0.710 ± 0.010
50.38%	49.62%	0.689 ± 0.010
90.38%	9.62%	0.661 ± 0.011

Completion-proxy progress is nonmonotone. Table 21 scores partial states after filling masked positions with the frozen model's deterministic argmax predictions. The held-out Enformer scores are averaged over 640 sequences per seed and then over five seeds. They are neither guidance reward R nor the batch-median Pred-Activity metric. D-DOIT's completion-proxy score rises from 2.067 at update 1 to 2.328 at update 32, falls to 1.758 at update 96, and reaches 3.670 at update 128; frozen sampling declines from 2.053 to 0.371. This trajectory shows higher endpoint activity without monotone intermediate progress. An argmax-completion score is not the rollout h-function or a guarantee about future reward.

Table 21: **Held-out enformer completion-proxy trajectory.** At each update, masked positions are filled with the frozen model's deterministic argmax predictions before scoring. Scores are averaged over 640 sequences per seed and then over five seeds, rather than using the batch-median Pred-Activity aggregation.

Update	1	32	64	96	128
D-DOIT	2.067	2.328	1.763	1.758	3.670
Frozen sampling	2.053	1.561	1.058	0.716	0.371

From update 32 onward, the argmax-completed D-DOIT sequences are 100% unique, and their mean pairwise Hamming distance increases from 0.726 to 0.749. This diversity measure concerns completed sequences across the sample batch, unlike the remaining-position conditional diversity above. The observed late-stage reward improvement is therefore compatible with maintained diversity in this diagnostic, rather than implying that guidance universally preserves diversity.

M Experimental Details of Protein Inverse Folding

Protein inverse folding tests amino-acid sequence design conditional on a target backbone, complementing the regulatory DNA experiments, and the full cross-method comparison, including all metrics and seed-level SDs, is in Table 22. We follow the protein inverse-folding protocol of DRAKES [8], using the same reward oracle and evaluation metrics. D-DOIT guides the frozen conditional generator through reward-value queries without updating its parameters.

Protocol and aggregation. We evaluate all 12 test protein backbones with three independent sampling seeds and generate 128 sequences per backbone and seed. The configuration $M = 52$, $K = 8$, and $n_h = 1$ (one completion per candidate) is fixed before test-set evaluation, with no tuning after inspecting test results. Each metric is computed separately for each seed under the DRAKES protocol; D-DOIT entries report the mean and sample standard deviation across those three seed-level metrics. The uncertainty describes sampling-seed variation, not variation across sequences or backbones.

Metrics. Pred-ddG is the predicted stability score, with higher values preferred under the benchmark's sign convention. We report its median and the percentage of sequences with predicted ddG greater than zero. Structural fidelity is measured by self-consistency RMSD (scRMSD), comparing the predicted structure of a generated sequence with its target backbone; we report the median in Å and the percentage below 2 Å. Joint success requires both positive predicted ddG and scRMSD below 2 Å. These are predictor-based measures rather than experimental stability or folding measurements.

Table 22: **Protein inverse folding on the DRAKES benchmark.** D-DOIT entries are mean ± sample SD across seed-level metrics. Other rows and uncertainties are from DRAKES Table 2 [8], not rerun. Bold marks the best mean per column, not statistical significance.

Method	Pred-ddG median ↑	ddG > 0 (%) ↑	scRMSD median (Å) ↓	scRMSD < 2 Å (%) ↑	Success (%) ↑
Pre-trained	-0.544 ± 0.037	36.6 ± 1.0	0.849 ± 0.013	90.9 ± 0.6	34.4 ± 0.5
CG	-0.561 ± 0.045	36.9 ± 1.1	0.839 ± 0.012	90.9 ± 0.6	34.7 ± 0.9
SMC	0.659 ± 0.044	68.5 ± 3.1	0.841 ± 0.006	93.8 ± 0.4	63.6 ± 4.0
TDS	0.674 ± 0.086	68.2 ± 2.4	$\mathbf{0.834 \pm 0.001}$	$\mathbf{94.4 \pm 1.2}$	62.9 ± 2.8
CFG	-1.186 ± 0.035	11.0 ± 0.4	3.146 ± 0.062	29.4 ± 1.0	1.3 ± 0.4
D-DOIT	$\mathbf{0.870 \pm 0.046}$	$\mathbf{74.8 \pm 3.1}$	0.872 ± 0.009	92.5 ± 1.3	$\mathbf{68.1 \pm 3.5}$

Results and comparison boundary. D-DOIT achieves the highest median Pred-ddG (0.870 ± 0.046), ddG-positive fraction ($74.8 \pm 3.1\%$), and joint success ($68.1 \pm 3.5\%$) among the six methods. Joint success exceeds the published SMC and TDS means by 4.5 and 5.2 percentage points, respectively. TDS retains better structural fidelity: median scRMSD is 0.834 Å versus 0.872 Å, and the fraction below 2 Å is 94.4% versus 92.5%. All non-D-DOIT entries are taken directly from DRAKES Table 2 and were not rerun in this comparison; their reported uncertainties are retained. The shared benchmark oracle and metrics permit a literature comparison, not a matched-compute or matched-implementation claim. The results extend the computational evaluation to protein sequence design; they do not establish experimentally measured stability or folding.

N Licenses for Existing Assets

We use pre-existing assets for non-commercial academic research and credit their creators through Sections J and K. The Masked Discrete Diffusion Model reference implementation of Sahoo et al. [3] carries Apache-2.0; the Ctrl-DNA repository of Chen et al. [25], Enformer implementation of Avsec et al. [64], and HyenaDNA repository of Nguyen et al. [55] also carry Apache-2.0 licenses. The HepG2 enhancer corpus from Gosai et al. [24] is distributed through the paper's data-availability resources, and the Nature article carries CC BY 4.0. The ENCODE Data Use Policy governs chromatin-accessibility data [65]; JASPAR 2022 [66] is open access and its Bioconductor data package is GPL-2. The public DRAKES repository associated with Wang et al. [8] had no top-level software license in the GitHub metadata at access time. We use DRAKES only as a cited benchmark, baseline, oracle, and checkpoint source for academic evaluation, and do not redistribute its code or checkpoints.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Section 1 state the method, its training-free scope, the Doob-transform construction, and the regulatory-DNA and protein inverse-folding evaluations. The claims are supported by experiments in Section 4 and limitations in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 5 includes a limitations paragraph describing dependence on the prediction-head proxy and the remaining scope for other discrete diffusion parameterizations and longer biological contexts.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The discrete h-transform underlying D-DOIT is stated as Theorem 2.1 in Section 2.2, with a complete proof in Section B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The algorithmic details are specified in Section 3 and the pseudocode, while datasets, reward oracles, evaluation protocols, hyperparameters, seed counts, and compute settings are reported in Sections J to M. These details provide a reproducibility path without assuming access to unreleased code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The public D-DOIT code artifact is available at https://github.com/WeiminWu2000/D-DOIT. It contains the benchmark adapters, generation entry points, environment requirements, and reproduction commands; existing datasets, checkpoints, and evaluation oracles are identified and cited in Sections J, K and M.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Sections J, K and M specify dataset sources, reward oracles, model/backbone setup, seed protocols, batch sizes, and evaluation metrics. Table 5 lists the sampler hyperparameters and their selected operating points.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: Main result tables report standard deviations across the stated random-seed protocols, including 3-seed DRAKES evaluation and 5-seed Ctrl-DNA evaluation, as specified in Sections J and K. Additional seed-level analyses and standard-deviation comparisons are reported in Section J.5; the paper does not interpret every numerical difference as statistically significant.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Sections J, K and M state the GPU type, batch sizes, seed counts, and inference protocols; the main rows use a single A100 80GB GPU, with the documented A100-SXM4-80GB endpoint control. D-DOIT is training-free and does not update the backbone during sampling. Section 4.4 and Section G report wall-clock and model-evaluation accounting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: To the best of the authors' knowledge, the research conforms to the NeurIPS Code of Ethics. The work uses cited public biological sequence benchmarks and pre-trained/evaluation assets and does not involve human-subject data collection.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: The current paper does not contain a separate broader-impact discussion covering both positive and negative societal impacts. It limits evaluation to predictive biological benchmarks and records asset and redistribution boundaries in Section N.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The current paper does not release a high-risk pre-trained foundation model, scraped dataset, or deployed biological design system. It reports benchmark experiments and does not redistribute the DRAKES code or checkpoints; no new model or dataset asset is released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: Existing datasets, checkpoints, reward oracles, baselines, and metric resources are credited through citations in Sections J, K and M. Section N lists the verified licenses and terms of use, and states the non-redistribution boundary for the DRAKES assets whose public repository does not declare a top-level software license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The public D-DOIT code artifact is documented and released at `https://github.com/a-green-hand-jack/ddoit-artifact`; it does not redistribute third-party datasets or model checkpoints. The method, pseudocode, hyperparameters, and benchmark protocols are documented in Sections 3, J, K and M and Table 5.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or new research with human subjects. It evaluates algorithms on existing regulatory DNA sequence design benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing or new research with human subjects, so IRB or equivalent human-subjects review is not applicable. The experiments use existing benchmark datasets and reward/evaluation oracles.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are not an important, original, or non-standard component of the core method, theory, experiments, data generation, evaluation, or result interpretation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.