ENHANCING PARALLEL TEMPERING FOR TEST-TIME STEERING IN MASKED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time steering aims to sample from reward-tilted distributions defined by a pre-trained model and an external reward function, without updating the model's parameters. However, samplers may struggle to move between high-reward regions separated by low-probability barriers, leaving important modes unexplored. Sequential Monte Carlo may concentrate particles around early-discovered modes, while replica exchange across diffusion noise levels may mix slowly when reward-agnostic exchange proposals are frequently rejected. We propose Temperature-Enhanced Masked Proposal and Exchange for Reward Steering (TEMPERS), a reward-centered parallel tempering framework for masked diffusion models that operates on complete sequences. Remask-and-denoise proposals enable reward-based acceptance at every local update, allowing each replica to refine candidates at its reward temperature before exchange. In an analytic example with a known reward-tilted distribution, TEMPERS closely approximates the target region probabilities. Experiments on protein and DNA sequence design and sentiment-controlled text generation further show higher rewards than the evaluated test-time steering baselines, alongside competitive output quality.

1 INTRODUCTION

Masked Diffusion Models (MDMs) provide expressive, high-fidelity priors for discrete sequence synthesis, with applications in DNA and protein design, text generation, and other structured sequences (de Groot et al., 2025; Shi et al., 2024; Nie et al., 2026). Many applications, however, need more than typical samples. They need several promising candidates that score well on a task-specific objective, for a costly downstream evaluation to select from. These requirements motivate test-time steering (Singhal et al., 2025; Yoon et al., 2025; Wang et al., 2026b): Given a pre-trained model $p(\boldsymbol{x})$ and an external reward $R(\boldsymbol{x})$, steering reshapes the prior toward a reward-tilted distribution without updating any model parameters:

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{Z} p(\boldsymbol{x}) \exp(\beta R(\boldsymbol{x})), \tag{1}$$

where β controls the strength of reward tilting (inverse temperature), and $Z := \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \exp(\beta R(\boldsymbol{x}))$ is the normalizing constant. Rather than returning only the highest-reward sequences, the goal is to cover the important regions of this distribution and return a set of candidates for downstream evaluation.

Sampling effectively from the reward-tilted distribution is challenging when high-reward regions are separated by low-probability barriers. Existing methods may mix slowly across these modes, causing the sampler to remain around early-discovered, suboptimal high-reward regions for many iterations. For SMC-based approaches (Wu et al., 2023), such as FK steering (Singhal et al., 2025), repeated reweighting and resampling further concentrate particles around suboptimal regions, potentially missing out other modes with even higher reward.

Parallel tempering, also known as replica exchange Markov chain Monte Carlo sampling, mitigates premature trapping by running multiple chains (replicas) with local updates at different temperatures and swapping between neighboring chains: higher-temperature chains explore across modes, while accepted swaps transfer newly discovered candidates toward lower-temperature chains for further refinement within high-reward regions. CREPE adapts this idea to diffusion models, by placing

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

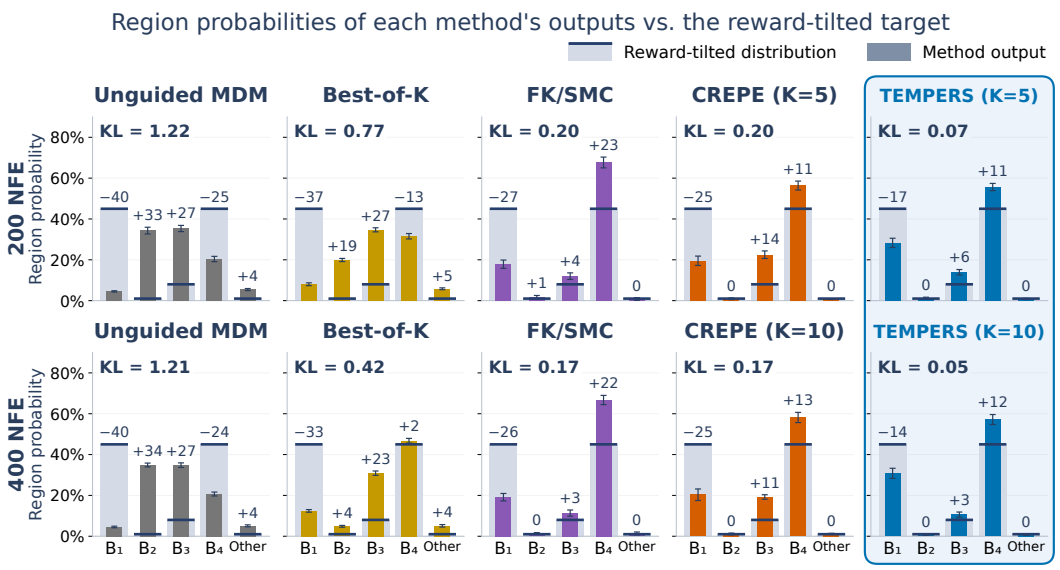


Figure 1: Probability mass for four-mode regions and others under 200- and 400-NFE budgets per run. TEMPERS and CREPE use $K = 5$ chains in the 200-NFE row and $K = 10$ in the 400-NFE row, with 8 warm-up updates in both rows. Actual costs are 200/400 NFE for TEMPERS and 198/396 NFE for CREPE in the two rows, respectively, including initialization. Results across 8 seeds and 256 runs per seed; error bars show the standard deviation across seed means. Numbers above the bars indicate the difference between the method and the target, in percentage points.

chains at different diffusion noise levels and using forward noising and backward denoising to generate proposals for exchange (He et al., 2026b). However, these proposals follow the pretrained diffusion dynamics without directly accounting for the reward and may therefore produce lower-reward candidates, leading to low acceptance rates and potentially slow mixing under reward tilting (see Figure 2 (C) in Section 4).

To address this limitation, we propose Temperature-Enhanced Masked Proposal and Exchange for Reward Steering (TEMPERS), a reward-centered parallel tempering framework that operates directly on complete sequences. Each replica operates at a different reward inverse temperature. To generate a new candidate, we leverage the remask-and-denoise process in MDMs (Ou et al., 2025; Shi et al., 2024): a fraction of the current sequence is masked and then regenerated by the MDM, producing a new complete sequence whose reward can be evaluated immediately. This allows the reward to enter the acceptance test for every local update, not just the exchange step, yielding candidates better aligned with the reward-tilted target. We further use larger remasking fractions at higher temperatures for broader exploration and smaller fractions at lower temperatures for local refinement. Together, these design choices aim to improve proposal efficiency and mixing across reward modes.

The synthetic experiments in Figure 1 demonstrate that our design yields more efficient proposals and faster mixing across reward modes, achieving the lowest Kullback–Leibler (KL) divergence to the reward-tilted target distribution among the compared methods under the same total number of function evaluations (NFE). Beyond these controlled experiments, we evaluate our method on protein and DNA sequence design, as well as sentiment-controlled text generation, obtaining substantial reward improvements while maintaining reasonable fidelity and diversity. See Section 4 for the full experimental setup and results.

2 Background

Masked Diffusion Models. Masked Diffusion Models (MDMs) define a generative model over discrete sequences $\boldsymbol{x} = (x_1, \ldots, x_L) \in \mathcal{V}^L$, where $\mathcal{V}$ is a finite vocabulary, i.e., the collection of all possible discrete tokens, and L is the sequence length. Unlike continuous diffusion models that inject

Gaussian noise, MDMs operate through *masking* (Sahoo et al., 2024). A special token [MASK] is introduced, and generation is learned by recovering complete sequences from partially masked inputs (Ou et al., 2025; Zheng et al., 2025).

Formally, let $\boldsymbol{x}^{(0)} \sim p_{\text{data}}$ denote a complete sequence. A forward corruption process constructs progressively noisier states $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$ via stochastic masking:

$$\boldsymbol{x}^{(t)} = \mathcal{M}_t(\boldsymbol{x}^{(0)}), \qquad t \in \{1, \dots, T\}, \tag{2}$$

where $\mathcal{M}_t$ increases the masking level with t. A standard formulation uses an increasing masking schedule $\sigma_t \in [0, 1]$ with $\sigma_T = 1$, where σ_t denotes the probability that a token is masked at step t (Sahoo et al., 2024; Nie et al., 2026). For each position i, we define

$$x_i^{(t)} = \begin{cases} [\text{MASK}] & \text{if } m_{t,i} = 1, \\ x_i^{(0)} & \text{otherwise,} \end{cases} \tag{3}$$

where

$$m_{t,i} \sim \text{Bernoulli}(\sigma_t). \tag{4}$$

This construction ensures that $\boldsymbol{x}^{(t)}$ preserves a subset of the original context while progressively increasing the proportion of masked tokens as t grows.

The reverse process is parameterized by a denoising model p_θ that predicts a complete sequence conditioned on the masked sequence:

$$p_\theta(\boldsymbol{x}^{(0)} \mid \boldsymbol{x}^{(t)}) = \prod_{i \in \mathcal{M}(\boldsymbol{x}^{(t)})} p_\theta(x_i^{(0)} \mid \boldsymbol{x}^{(t)}), \tag{5}$$

where $\mathcal{M}(\boldsymbol{x}^{(t)})$ denotes the masked positions. Training minimizes the expected cross-entropy over masked tokens (Ou et al., 2025):

$$\mathcal{L}_{\text{MDM}}(\theta) = \mathbb{E}_{t, \boldsymbol{x}^{(0)}, \boldsymbol{x}^{(t)}} \left[- \sum_{i \in \mathcal{M}(\boldsymbol{x}^{(t)})} \log p_\theta(x_i^{(0)} \mid \boldsymbol{x}^{(t)}) \right]. \tag{6}$$

Remask-and-Denoise Generation. Sampling from an MDM p_θ is started from a fully masked sequence $\boldsymbol{x}_{\text{full}} = [\text{MASK}, \dots, \text{MASK}]$, with the initial state sampled as $\hat{\boldsymbol{x}}^{(T)} \sim p_\theta(\cdot \mid \boldsymbol{x}_{\text{full}})$.

Generation then proceeds backward from $t = T$ to 1 through an iterative remask-and-denoise process (Nie et al., 2026; Wang et al., 2026a). Given the current state $\hat{\boldsymbol{x}}^{(t)}$, a remasking operator first produces a partially masked intermediate state:

$$\hat{\boldsymbol{x}}_{\text{mask}}^{(t)} \sim \rho(\cdot \mid \hat{\boldsymbol{x}}^{(t)}, m), \tag{7}$$

where ρ selects $\lfloor mL \rfloor$ tokens to reset to [MASK] and $m \in [0, 1]$ controls the expected proportion of tokens to be remasked. In this paper, we consider the uniform remasking, where we first sample a subset:

$$\mathcal{S} \sim \text{Unif}\left(\{S \subseteq \{1, 2, 3, \dots, L\} \mid |S| = \lfloor mL \rfloor\}\right), \tag{8}$$

and construct the partially masked sequence $\hat{\boldsymbol{x}}_{\text{mask}}^{(t)}$ by

$$\hat{\boldsymbol{x}}_{\text{mask},i}^{(t)} = \begin{cases} [\text{MASK}], & i \in \mathcal{S}, \\ \hat{\boldsymbol{x}}_{\text{mask},i}^{(t)} & i \notin \mathcal{S}. \end{cases} \tag{9}$$

Conditioned on this input, the denoiser samples the next state:

$$\hat{\boldsymbol{x}}^{(t-1)} \sim p_\theta(\cdot \mid \hat{\boldsymbol{x}}_{\text{mask}}^{(t)}). \tag{10}$$

Together, these define the one-step transition kernel

$$g_\theta(\hat{\boldsymbol{x}}^{(t-1)} \mid \hat{\boldsymbol{x}}^{(t)}) = \sum_{\boldsymbol{x}_{\text{mask}}^{(t)}} p_\theta(\hat{\boldsymbol{x}}^{(t-1)} \mid \hat{\boldsymbol{x}}_{\text{mask}}^{(t)}) \rho(\hat{\boldsymbol{x}}_{\text{mask}}^{(t)} \mid \hat{\boldsymbol{x}}^{(t)}, m). \tag{11}$$

Notice that the generated states $\{\hat{\boldsymbol{x}}^{(t)}\}_{t=0}^{T}$ are complete sequences except in implementations that allow residual mask tokens at intermediate steps. In typical implementations, the final output $\hat{\boldsymbol{x}}^{(0)}$ is required to be a complete sequence with no remaining mask tokens.

Algorithm 1 Discrete Temperature-Enhanced Masked Proposal and Exchange for Reward Steering

Input: pre-trained MDM p_θ, reward $R(\boldsymbol{x})$, K chains with ladders $0 \le \beta_1 < \cdots < \beta_K = \beta$ and $m_1 > \cdots > m_K$, iterations N, exchange interval τ_{ex}.
Initialize: $\boldsymbol{x}_0^{[k]} \sim p_\theta$ from fully masked sequences, for $k = 1, \ldots, K$.
for $n = 1$ to N **do**
  for $k = 1$ to K **do** ▷ Within-chain update
    Remask a fraction m_k of $\boldsymbol{x}_{n-1}^{[k]}$ into $\boldsymbol{x}_{\text{mask}}^{[k]}$ and propose $\tilde{\boldsymbol{x}}^{[k]} \sim p_\theta(\cdot \mid \boldsymbol{x}_{\text{mask}}^{[k]})$.
    With probability A_k in Eq. (16), set $\boldsymbol{x}_n^{[k]} \leftarrow \tilde{\boldsymbol{x}}^{[k]}$; otherwise $\boldsymbol{x}_n^{[k]} \leftarrow \boldsymbol{x}_{n-1}^{[k]}$.
  if $n \bmod \tau_{\text{ex}} = 0$ **then** ▷ replica exchange
    for $k = 1$ to $K - 1$ **do**
      Exchange $\boldsymbol{x}_n^{[k]} \leftrightarrow \boldsymbol{x}_n^{[k+1]}$ with probability A_{ex} in Eq. (20).
Output: Final steered sample $\boldsymbol{x}_{\text{PT}} = \boldsymbol{x}_N^{[K]}$ from the coldest chain.

3 TEMPERS: TEMPERATURE-ENHANCED MASKED PROPOSAL AND EXCHANGE FOR REWARD STEERING

TEMPERS leverages the remask-and-denoise proposal mechanism to iteratively generate new candidates by refining the complete sequence. Starting from the complete sequence, TEMPERS maintains a set of interacting chains operating at different reward temperatures by remasking at a random position with a different masking ratio. Then, multiple chains are run with different reward intensities: each chain holds a complete sequence as its current state, corresponding to a different reward-tilted distribution. High-temperature chains are paired with larger modification scales for broad exploration, while low-temperature chains are paired with smaller scales for local refinement. After each chain generates a proposal, acceptance is determined by the reward and the chain's temperature, enabling within-chain updates to favor high-reward candidates.

Replica exchange is then used to allow states to flow between different temperatures. This allows for separate control and coordination of reward preference and proposal scale, ensuring reward feedback participates in each round of within-chain updates and linking exploration and refinement through exchange. The full procedure is summarized in Algorithm 1.

Temperature ladder and target distributions. We maintain K chains indexed by increasing inverse temperatures

$$0 \le \beta_1 < \beta_2 < \cdots < \beta_K = \beta, \tag{12}$$

where $\beta_K = \beta$ is the user-specified target reward temperature of interest. Each chain targets the tempered distribution

$$p_{\beta_k}(\boldsymbol{x}) = \frac{1}{Z_{\beta_k}} p_\theta(\boldsymbol{x}) \exp(\beta_k R(\boldsymbol{x})). \tag{13}$$

In particular, small β_k favors exploration under the prior, while large β_k increasingly emphasizes reward preference.

Local updates via temperature-scaled remasking. To construct local proposals within each chain, we adapt the standard remask-and-denoise procedure (Section 2) by introducing temperature-scaled masking fractions $\{m_k\}_{k=1}^K$ such that $1 \ge m_1 > m_2 > \cdots > m_K > 0$.

Given the current state $\boldsymbol{x}^{[k]}$ in chain k, a proposal is generated via:

$$\boldsymbol{x}_{\text{mask}}^{[k]} \sim \rho(\cdot \mid \boldsymbol{x}^{[k]}, m_k), \qquad \tilde{\boldsymbol{x}}^{[k]} \sim p_\theta(\cdot \mid \boldsymbol{x}_{\text{mask}}^{[k]}), \tag{14}$$

where ρ remasks exactly $\lfloor m_k L \rfloor$ tokens of the length-L sequence.

Within-chain acceptance. We decide whether to accept the proposal $\tilde{\boldsymbol{x}}^{[k]}$ as the next state of chain k using a Metropolis-Hastings update. Write $\Delta R_k := R(\tilde{\boldsymbol{x}}^{[k]}) - R(\boldsymbol{x}^{[k]})$. The exact acceptance

probability is

$$A_k^{\mathrm{MH}}(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]}) = \min\left\{1, \frac{p_\theta(\tilde{\boldsymbol{x}}^{[k]})}{p_\theta(\boldsymbol{x}^{[k]})} \frac{g_k(\boldsymbol{x}^{[k]} \mid \tilde{\boldsymbol{x}}^{[k]})}{g_k(\tilde{\boldsymbol{x}}^{[k]} \mid \boldsymbol{x}^{[k]})} \exp(\beta_k \Delta R_k)\right\}. \tag{15}$$

However, for masked diffusion proposals, the model ratio p_θ is intractable and the proposal density $g_k(\boldsymbol{x}^{[k]} \mid \tilde{\boldsymbol{x}}^{[k]})$ is computationally expensive, as it requires marginalizing over all possible masking configurations. We instead use a proxy that looks at nothing but the reward difference:

$$A_k(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]}) = \min\left\{1, \exp(\beta_k \Delta R_k)\right\}. \tag{16}$$

In particular, under the uniform remasking scheme described in Section 2, suppose that masked tokens are unmasked sequentially, with each token sampled from the exact conditional distribution under the same prior p_θ used in the target, conditioned on the unmasked context and all previously regenerated tokens. Then the exact Metropolis–Hastings acceptance probability coincides with the reward-only proxy:

Proposition 1. *Assume the remasking distribution ρ selects a mask of fixed size $\lfloor m_k L \rfloor$ uniformly at random. Then the exact Metropolis-Hastings acceptance probability $A_k^{\mathrm{MH}}(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]})$ for the remask-and-denoise proposal kernel g_k simplifies exactly to:*

$$A_k^{\mathrm{MH}}(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]}) = A_k(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]}) = \min\left\{1, \exp(\beta_k \Delta R_k)\right\}. \tag{17}$$

The proof is provided in Appendix A. This result provides a sufficient condition under which our simplified acceptance rule yields a transition kernel satisfying exact detailed balance with respect to the reward-tilted distribution.

Remark. In the main experiments reported in Section 4, we use the denoiser to unmask all masked tokens jointly, completing the masked positions in a single denoising step before evaluating the reward on the resulting sequence. This procedure is similar in spirit to Tweedie-based intermediate reward estimation (Singhal et al., 2025; Li et al., 2025), where one first predicts a clean sample and then evaluates its reward. We show that this one-step denoising strategy already achieves reasonable empirical performance. In Appendix C.5, we further increase the number of denoising steps, bringing the practical procedure closer to the sequential-denoising assumption used in the analysis above.

Replica exchange for global exploration. To enable communication across chains, we periodically attempt exchanges between adjacent chains. This mechanism allows high-reward candidates discovered by exploratory (low-β) chains to propagate to exploitative (high-β) chains, while those same exploitative chains receive diverse states that help them escape local modes. For chains k and $k+1$, we propose the exchange

$$(\boldsymbol{x}^{[k]}, \boldsymbol{x}^{[k+1]}) \to (\boldsymbol{x}^{[k+1]}, \boldsymbol{x}^{[k]}), \tag{18}$$

accepted with probability A_{ex} given below. Let $\Delta R_{k,k+1} := R(\boldsymbol{x}^{[k]}) - R(\boldsymbol{x}^{[k+1]})$.

$$A_{\mathrm{ex}} = \min\left\{1, \frac{p_{\beta_k}(\boldsymbol{x}^{[k+1]}) p_{\beta_{k+1}}(\boldsymbol{x}^{[k]})}{p_{\beta_k}(\boldsymbol{x}^{[k]}) p_{\beta_{k+1}}(\boldsymbol{x}^{[k+1]})}\right\}. \tag{19}$$

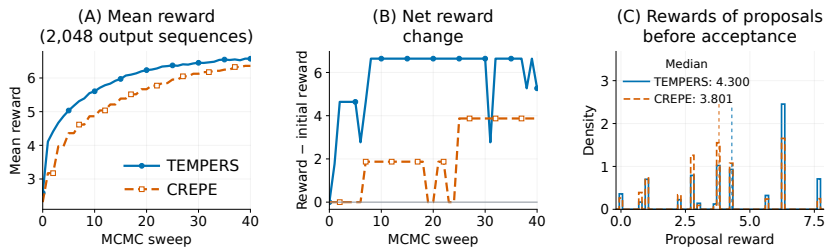
Because the shared prior cancels, this simplifies to

$$A_{\mathrm{ex}} = \min\left\{1, \exp\left((\beta_{k+1} - \beta_k)\Delta R_{k,k+1}\right)\right\}. \tag{20}$$

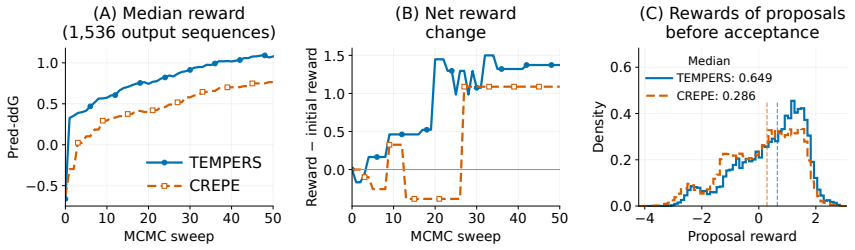
Especially, the prior p_θ cancels exactly in the ratio, so the exchange criterion depends only on reward differences, making it both theoretically exact and cheap to evaluate in practice.

4 Experiments

We first evaluate TEMPERS on a synthetic sequence dataset with a multimodal reward landscape to verify its exploration capability. We then apply TEMPERS to representative benchmarks, including inverse protein folding, regulatory DNA sequence design, and sentiment-controlled text generation.



(a) Analytic experiments ($K = 5$).



(b) Protein sequence design.

Figure 2: **Reward progression and exchange proposals for TEMPERS and CREPE.** (a) Analytic experiments with 2,048 sequences over 40 sweeps; (b) protein sequence design with 1,536 sequences over 50 sweeps. Within each subfigure, (A) shows the mean (analytic) or median (protein) reward of the current output sequences; (B) shows reward minus its initial value for selected individual systems; and (C) shows the rewards of all complete exchange proposals before acceptance, including rejected proposals. Histograms use 64 shared bins per task and integrate to one per method; dashed lines mark medians.

4.1 Analytic Examples

The reward-tilted distribution is unavailable in real-world scenarios, so we design a prior and a multimodal reward landscape over the discrete sequence space to conduct analysis, focusing on four mode regions, $B_1, \ldots, B_4$. For the prior, B_1 is the region with low prior density but high reward samples; B_2 is the valley of the reward landscape where the sampler cannot obtain positive rewards; B_3 is the high mode in the prior but low in the reward-tilted distribution; B_4 is the second high mode and increases after allocating high reward landscape as shown in Appendix Figure 4. This setup includes different relationships between the prior and the reward-tilted distribution, which helps us verify the characteristics of different sampling methods.

We compare TEMPERS against CREPE, FK Steering (FK/SMC), and Best-of-K for all experiments in our paper. Figure 1 shows that FK/SMC fails in the local-mode trap, highlighting its limited ability to explore rare-event regions with low prior probability mass but high reward. In contrast, Parallel Tempering-based methods such as TEMPERS and CREPE sample less data than FK/SMC in region B_4. Additionally, we observe that CREPE is less effective than TEMPERS for small NFEs, indicating that different designs yield a gap. We compare TEMPERS and CREPE in Section 4.3. Appendix C.1 reports the detailed experimental setup and results.

4.2 Representative Benchmarks

4.2.1 DNA Sequence Design

In this section, we reuse the masked diffusion model pre-trained on the large-scale DNA enhancer dataset (Wang et al., 2025; Lee et al., 2026), which captures the thousands of unique cell types in the body. The reward function (Activity) evaluates the enhancer activity level of a generated DNA sequence in a specific cell type, such as the HepG2 cell line. We also evaluate sequence naturalness using 3-mer Pearson Correlation (3-mer Corr) and measure biological validity with an independent binary classifier (ATAC-Acc) trained on chromatin accessibility data in the HepG2 cell line (ENCODE Project Consortium et al., 2012). Unlike the analytic experiment in Section 4.1 with a ground-truth reward-tilted distribution, we need to evaluate with multiple metrics to assess a method's effectiveness.

Table 1: (DNA sequence design) Model performance on regulatory DNA sequence design. Method* denotes our implementation or reproduction.

Group	Method	Activity ↑	ATAC-Acc (%) ↑	3-mer Corr ↑	Diversity (%) ↑
Reference	Pretrained*	0.181 ± 0.011	1.8 ± 0.4	-0.058 ± 0.026	74.65 ± 0.06
Test-time guidance	CG	3.30 ± 0.00	0.0 ± 0.0	-0.065 ± 0.001	–
	SMC	4.15 ± 0.33	39.9 ± 8.7	0.840 ± 0.045	–
	TDS	4.64 ± 0.21	45.3 ± 16.4	0.848 ± 0.008	–
	Best-of-K (K=10)*	1.787 ± 0.077	7.6 ± 1.0	0.530 ± 0.033	**74.82 ± 0.01**
	FK-Steering (K=10)*	5.047 ± 0.012	35.7 ± 1.6	0.838 ± 0.005	73.52 ± 0.04
	CREPE (K=10)*	5.274 ± 0.063	50.9 ± 1.6	0.861 ± 0.015	74.21 ± 0.04
Ours	TEMPERS (K=10)*	**5.466 ± 0.022**	**61.2 ± 1.4**	**0.918 ± 0.013**	74.13 ± 0.03

Table 2: (Protein sequence design) Model performance on inverse protein folding. Method* denotes our implementation, and other methods are reported as reference results from DRAKES (Wang et al., 2025). We report its mean ± standard deviation across three seeds.

Group	Method	DDG ↑	%(DDG> 0) ↑	scRMSD ↓	%(scRMSD< 2) ↑	Success Rate ↑	Diversity (%) ↑
Reference	Pretrained*	-0.544 ± 0.037	36.6 ± 1.0	0.849 ± 0.013	90.9 ± 0.6	34.4 ± 0.5	24.1 ± 0.1
Test-time guidance	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	**0.834 ± 0.001**	**94.4 ± 1.2**	62.9 ± 2.8	–
	Best-of-K (K=10)*	0.348 ± 0.011	59.0 ± 0.4	0.869 ± 0.006	91.7 ± 0.4	54.1 ± 0.4	23.8 ± 0.1
	FK-Steering (K=10)*	0.704 ± 0.021	69.6 ± 0.5	0.857 ± 0.011	93.9 ± 0.4	64.5 ± 0.4	21.4 ± 0.1
	CREPE (K=10)*	0.752 ± 0.035	69.9 ± 1.1	0.871 ± 0.005	91.1 ± 0.3	63.3 ± 1.2	29.1 ± 0.1
Ours	TEMPERS (K=10)*	**1.066 ± 0.007**	**77.6 ± 0.8**	0.880 ± 0.006	91.7 ± 0.5	**70.3 ± 0.7**	**29.5 ± 0.1**

As shown in Table 1, TEMPERS outperforms test-time steering baselines in generating enhancer DNA sequences with the highest rewards (Activity). In addition to achieving the highest rewards, TEMPERS maintains natural, biologically valid sequences with the highest 3-mer Corr and ATAC-Acc. Appendix C.3 reports the detailed experimental setup and results.

4.2.2 Protein sequence design

This task aims to steer the pre-trained masked diffusion model (as the inverse folding model) to generate amino acid sequences toward high stability (as the reward function) (Wang et al., 2025; Widatalla et al., 2026). Specifically, we generate 1536 sequences conditioned on 12 structures selected from the Megascale dataset (Tsuboyama et al., 2023). The reward function measures a protein sequence's folding stability using Delta Delta G (DDG), which predicts how a single point mutation affects protein stability.[1] We also evaluate whether a generated amino acid sequence folds into a conditional structure using self-consistency RMSD (scRMSD), which uses ESMFold (Lin et al., 2023) to predict the sequence structure and calculate RMSD relative to the original sequence's structures in Megascale (Chu et al., 2024; Wang et al., 2025). We follow the experimental protocol in DRAKES (Wang et al., 2025), which defines the inverse-folding success rate using two criteria: DDG > 0 and scRMSD < 2.

Table 2 reports the median DDG and scRMSD, and applies both criteria to obtain the overall success rate. TEMPERS generates amino acid sequences with higher DDG than the evaluated test-time steering methods, exceeding the strongest baseline by 0.3 and the pre-trained masked diffusion model by 1.6. Meanwhile, TEMPERS maintains a reasonable scRMSD while generating the desired structure and achieves the highest Success Rate (70%). These results suggest that TEMPERS explores the reward landscape more effectively for identifying stable protein sequences than existing test-time steering approaches do under the same conditional structures. The results also show higher output diversity for TEMPERS and CREPE than for Best-of-K and FK Steering (FK/SMC), with the same ranking across all three seeds. Appendix C.2 reports the detailed experimental setup and results.

[1] More information of DDG: `https://support.levitate.bio/bench/DDG/`

Table 3: We evaluate LLaDA-8B-Instruct on 60 prompts from IFEval under both positive and negative sentiment steering, yielding 120 instances. Results reported as mean ± standard error. **Sentiment** is the classifier score aligned to the target polarity, obtained by negating the score for negative-target generations; higher is better. **Strict** and **Loose** are IFEval prompt-level accuracies, **PPL** are perplexity.

Group	Method	Sentiment ↑	Strict ↑	Loose ↑	PPL ↓
Reference	Pretrained	0.000 ± 0.087	0.517 ± 0.046	0.517 ± 0.046	4.7 ± 0.4
Test-time steering	Best-of-K (K=10)	0.600 ± 0.070	0.467 ± 0.046	0.525 ± 0.046	$\mathbf{4.2 \pm 0.2}$
	SMC	0.686 ± 0.050	0.358 ± 0.044	0.383 ± 0.045	246.1 ± 26.6
	CREPE (K=10)	0.703 ± 0.062	0.483 ± 0.046	0.525 ± 0.046	22.7 ± 3.3
Ours	TEMPERS (K=10)	$\mathbf{0.734 \pm 0.057}$	$\mathbf{0.575 \pm 0.045}$	$\mathbf{0.583 \pm 0.045}$	6.5 ± 1.3

4.2.3 TEXT SENTIMENT CONTROL

In this task, we steer masked language diffusion models (MLDMs) to generate sentences with positive or negative sentiment. Specifically, we generate sentences by prompting the LLaDA-8B-Instruct pre-trained model and steer them using a fine-tuned DistilBERT as the sentiment reward. We use IFEval (Instruction-Following Eval) to evaluate instruction-following compliance under sentiment steering (Zhou et al., 2023) and report the perplexity (PPL) that LLaDA-8B-Instruct itself assigns to each response, a fluency measure that does not involve the reward. Since sentiment reward alone cannot distinguish genuine steering from over-optimization, we read it alongside IFEval accuracy and PPL, which are independent of the reward.

Table 3 shows that, given a sentiment classifier, TEMPERS steers the pre-trained LLaDA-8B-Instruct to generate text with positive or negative sentiment across scenarios such as an all-capital answer about *praying for someone's health*, a professional *joke about morphology*, and a two-scene *dialogue that avoids the letter t* (Table 5). Meanwhile, TEMPERS achieves a reasonable PPL (< 100) relative to SMC's steering results (Ou et al., 2026) and still maintains instruction-following capability across strict and loose sentence structures. Appendix C.4 reports the settings and cost of every method and the responses before and after steering.

4.3 ABLATION STUDY FOR TEMPERS

Figure 1 compares the performance of TEMPERS and CREPE. This section further analyzes the performance gap by comparing changes in reward across MCMC sweeps. Next, we examine the impact of single-chain versus multiple-chain runs on the Analytic Experiments. Moreover, we analyze the sensitivity of the inverse temperature and re-masking rate steps to protein design results for the practical use of TEMPERS.

TEMPERS vs CREPE Figure 2 shows the comparisons between TEMPERS and CREPE in Analytic Examples and Protein Sequence Design. The aggregate output reward of TEMPERS generally exceeded that of CREPE during sampling and reached high values in the early sweeps. In the selected individual MCMC trajectories, the net reward change shows longer early plateaus for CREPE. When examining candidate sequences prior to acceptance, the reward distribution for TEMPERS proposals was skewed toward the higher end, with a relatively high median. The experimental analysis reveals that TEMPERS offers high-reward proposals and exchanges states across chains while keeping high rewards.

Number of chains Figure 3 shows that increasing the number of chains makes the output distribution of TEMPERS closer to the reward-tilted target distribution. With the same 200-NFE budget, increasing K from 1 to 5 reduces KL from 1.15 to 0.07, and the previously higher probabilities in B_2 and B_3 also decrease. At $K = 10$, using 400 NFE, KL further decreases to 0.05. The $K = 5$ and $K = 10$ runs both perform 20 updates, so this last comparison increases the model-call budget along with the number of chains. The ablation study reveals that TEMPERS leverages multiple chains at different temperatures to enable effective exploration, outperforming single-chain updating with the Metropolis-Hastings algorithm on complete sequences (Phunyaphibarn & Sung, 2026).

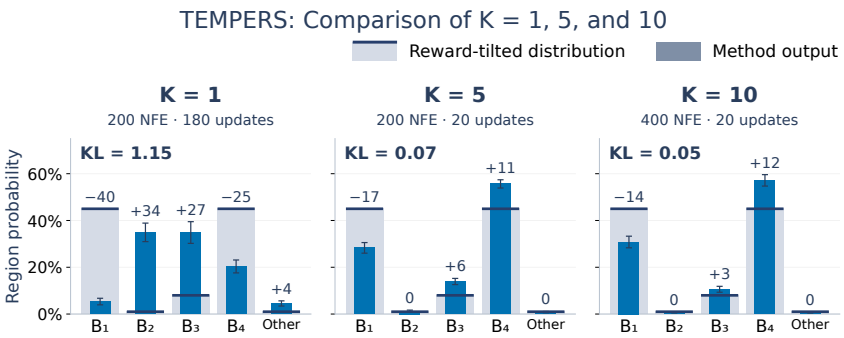


Figure 3: TEMPERS output distributions with $K = 1$, 5, and 10 chains. The $K = 1$ and $K = 5$ runs use 200 NFE with 180 and 20 updates, respectively; $K = 10$ uses 400 NFE with 20 updates. Costs include initialization and 8 warm-up updates. Results run across 8 seeds and 256 runs per seed; error bars show the standard deviation across seed means. Numbers above bars indicate method minus target in percentage points.

5 Related work

Test-time steering for diffusion models. Test-time steering for diffusion models is commonly formulated as sampling from a reward-tilted distribution that balances the pre-trained prior with an external reward signal. This view has motivated a broad family of test-time methods that keep the pre-trained model fixed while spending additional inference compute: Best-of-K selection, classifier or reward guidance, and Sequential Monte Carlo (SMC) methods such as tree-search steering, and Feynman–Kac (FK) Steering (Mudgal et al., 2024; Schiff et al., 2025; Guo et al., 2026; Jain et al., 2026; Wu et al., 2023; Hasan et al., 2026; He et al., 2026a). These methods avoid reward-specific retraining but face a finite-budget sampling problem. SMC and FK Steering allocate computation via weighting and resampling based on intermediate reward estimates (Singhal et al., 2025). With few particles and unreliable early estimates, this may discard particles whose potential becomes apparent only later, motivating careful use of intermediate signals. TEMPERS, on the other hand, adopts replica exchange, where candidates found by high-temperature chains can be passed to low-temperature chains, allowing exploration and improvement to complement each other.

Parallel tempering for diffusion models Finite-particle guidance can over-optimize within a mode and misweight distant high-reward modes (Dandapanthula & Boffi, 2026). Parallel tempering addresses this exploration-exploitation trade-off by maintaining interacting chains at multiple temperatures (Earl & Deem, 2005). Related tempering methods include source parallel tempering in the source space of flow-based generators (Wang et al., 2026b), PATHS for reward-aware SMC initialization (Oh et al., 2026), and PT-based global initialization followed by conditional diffusion refinement (Castro-Macías et al., 2026). These works study continuous diffusion models for image generation or Protein tertiary structure generation, rather than discrete diffusion models. See Appendix B for extended discussion.

6 Conclusion

This study proposes TEMPERS, a masked diffusion model test-time steering method that does not require updating the pre-trained model parameters. Through remask-and-denoise proposals, TEMPERS incorporates reward feedback at each local update and coordinates broad exploration with local refinement by using different reward temperatures, remasking ratios, and replica exchange. Analytical experiments show that TEMPERS more accurately approximates the probability in reward-skewed regions of the target distribution. Results on protein and DNA sequence design, as well as text sentiment control, also demonstrate that its reward performance outperforms comparable test-time steering methods while maintaining output quality. These results support combining reward feedback on the complete sequence with parallel tempering as a feasible direction to improve the effectiveness of discrete test-time steering. Future research could explore adaptive configurations of temperature and remasking ratios to improve sampling efficiency under limited computational budgets.

AI DISCLOSURE

In this work, we used generative AI tools to support research ideation and execution, assist with literature retrieval and the discovery of related work, draft sections of the manuscript, and revise and polish the writing. The authors retain responsibility for verifying the accuracy of AI-assisted text, references, methods, and results, and take full responsibility for the final content of this paper, including all claims and artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

To support reproducibility, Appendix C provides detailed experimental descriptions for the analytic examples, protein sequence design, regulatory DNA sequence design, and text sentiment control, including the datasets, pretrained models, reward functions, sampling configurations, and evaluation protocols. It also describes the additional ablation studies. Section 3 and Algorithm 1 specify the proposed sampling procedure.

REFERENCES

Eden Avrahami and Eliya Nachmani. Ilrr: Inference-time steering method for masked diffusion language models. *arXiv preprint arXiv:2601.21647*, 2026.

Francisco M. Castro-Macías, Pablo Morales-Álvarez, Saifuddin Syed, Daniel Hernández-Lobato, Rafael Molina, and José Miguel Hernández-Lobato. Conditional diffusion sampling. *arXiv preprint arXiv:2605.04013*, 2026. URL https://arxiv.org/abs/2605.04013.

Alexander E. Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W. Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024. doi: 10.1073/pnas.2311500121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2311500121.

Sanjit Dandapanthula and Nicholas M. Boffi. Are we really tilting? the mechanics of reward guidance in flow and diffusion models. *arXiv preprint arXiv:2606.02884*, 2026. URL https://arxiv.org/abs/2606.02884.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://www.science.org/doi/abs/10.1126/science.add2187.

Lars de Groot, Ruurd Jan Anthonius Kuiper, and Ayoub Bagheri. A survey of discrete diffusion for text and genomic sequence generation. In *The 37th Benelux Conference on Artificial Intelligence and the 34th Belgian Dutch Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ubOzVt7oMw.

David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

a) ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57, 2012.

Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. *Advances in Neural Information Processing Systems*, 38:73343–73384, 2026.

Mohsin Hasan, Viktor Ohanesian, Artem Gazizov, Yoshua Bengio, Alán Aspuru-Guzik, Roberto Bondesan, Marta Skreta, and Kirill Neklyudov. Discrete feynman-kac correctors. *arXiv preprint arXiv:2601.10403*, 2026.

Jiajun He, José Miguel Hernández-Lobato, Yuanqi Du, and Francisco Vargas. Rne: Plug-and-play diffusion inference-time control and energy-based training. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7a99ad21706dec5b28f9ad715e12197f-Abstract-Conference.html.

Jiajun He, Paul Jeha, Peter Potaptchik, Leo Zhang, José Miguel Hernández-Lobato, Yuanqi Du, Saifuddin Syed, and Francisco Vargas. CREPE: Controlling diffusion with REPlica exchange. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=uOZRWcbiZl.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *Advances in Neural Information Processing Systems*, 38:145576–145615, 2026.

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.

Jaewoo Lee, Minsu Kim, Sanghyeok Choi, Inhyuck Song, Sujin Yun, Hyeongyu Kang, Woocheol Shin, Taeyoung Yun, Kiyoung Om, and Jinkyoo Park. Diffusion alignment as variational expectation-maximization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=aBeIFDshvZ.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6QbbaEGkO7.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=bVIcZb7Qa0.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=KnqiC0znVF.

Myeongjun Oh, Gwangho Kim, and Sungyoon Lee. Parallel tempering initial sampling in inference-time reward alignment. *arXiv preprint arXiv:2605.30991*, 2026.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Zijing Ou, Chinmay Pani, and Yingzhen Li. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7wbrFQvfdH.

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. URL https://arxiv.org/abs/2602.09424v1.

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*, volume 2025, pp. 25383–25414, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i5MrJ6g5G1.

Yair Schiff, Omer Belhasin, Roy Uziel, Guanghan Wang, Marianne Arriola, Gilad Turok, Michael Elad, and Volodymyr Kuleshov. Learn from your mistakes: Self-correcting masked diffusion models, 2026. URL https://arxiv.org/abs/2602.11590.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

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 *ICML*, 2025. URL https://proceedings.mlr.press/v267/singhal25b.html.

Robert H. Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57:2607–2609, Nov 1986. doi: 10.1103/PhysRevLett.57.2607. URL https://link.aps.org/doi/10.1103/PhysRevLett.57.2607.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, and Gabriel J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444, 2023. doi: 10.1038/s41586-023-06328-6. URL https://doi.org/10.1038/s41586-023-06328-6.

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 *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9qzpNSTUYp.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G328D1xt4W.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026a. URL https://openreview.net/forum?id=IJryQAOy0p.

Shih-Hsin Wang, Jordan Keller, Taylor Transue, Dustin Brown, Thomas Strohmer, and Bao Wang. Test-time guidance for flow-based generative models via parallel tempering on source distributions. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/pdf?id=5jGcYMUZx1.

Talal Widatalla, Ashir A. Borah, Samuel H. King, Claudia L. Driscoll, Rafael Rafailov, and Brian L. Hie. Aligning protein-generative models to experimental fitness with proteindpo. *Nature Methods*, 23(9):1805–1813, 2026. doi: 10.1038/s41592-026-03137-3. URL https://doi.org/10.1038/s41592-026-03137-3.

Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=eWKqr1zcRv.

TaeHoon Yoon, Yunhong Min, Kyeongmin Yeo, and Minhyuk Sung. ψ-sampler: Initial particle sampling for SMC-based inference-time reward alignment in score models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=slVqJAI5sT.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CTC7CmirNr.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Proof of Proposition 1

Proof. Let $\boldsymbol{x}^{[k]} = (x_1^{[k]}, \dots, x_L^{[k]})$ and $\tilde{\boldsymbol{x}}^{[k]} = (\tilde{x}_1^{[k]}, \dots, \tilde{x}_L^{[k]})$ denote the current and proposed discrete sequences of length L, respectively. For any subset of indices $A \subseteq \{1, \dots, L\}$, we let $\boldsymbol{x}_A^{[k]}$ denote the sub-sequence of tokens at the positions specified by A, and let $\boldsymbol{x}_{\setminus A}^{[k]}$ denote the unmasked context, namely the sub-sequence of tokens at the complementary positions $\{1, \dots, L\} \setminus A$.

Let $M \subset \{1, \dots, L\}$ denote the set of indices chosen to be masked. Under the assumption of uniform remasking with a fixed masking fraction m_k, the size of the mask is strictly defined as $|M| = \lfloor m_k L \rfloor$. The probability of selecting any specific mask M of this size is a combinatorial constant independent of the actual token values in the sequence:

$$\rho(\boldsymbol{x}_{\setminus M}^{[k]} \mid \boldsymbol{x}^{[k]}, m_k) = \rho(\boldsymbol{x}_{\setminus M}^{[k]} \mid \tilde{\boldsymbol{x}}^{[k]}, m_k) = \frac{1}{\binom{L}{\lfloor m_k L \rfloor}} \equiv C, \tag{21}$$

where we slightly abuse notation and use $\boldsymbol{x}_{\setminus M}^{[k]}$ to denote the remasked version of $\boldsymbol{x}^{[k]}$.

Next, let $D = \{i \in \{1, \dots, L\} : x_i^{[k]} \neq \tilde{x}_i^{[k]}\}$ denote the set of indices where the current and proposed sequences differ. For the forward proposal kernel $g_k(\tilde{\boldsymbol{x}}^{[k]} \mid \boldsymbol{x}^{[k]})$ to have a non-zero probability of generating $\tilde{\boldsymbol{x}}^{[k]}$, the sampled mask M must cover all differing tokens. Otherwise, the unmasked tokens would prevent the transition. Therefore, we require $D \subseteq M$. Consequently, for any valid mask M, the unmasked context remains identical between the two sequences: $\boldsymbol{x}_{\setminus M}^{[k]} = \tilde{\boldsymbol{x}}_{\setminus M}^{[k]}$.

Using this property, we can write the forward proposal kernel as a sum over all valid masks that contain the differing tokens:

$$\begin{aligned} g_k(\tilde{\boldsymbol{x}}^{[k]} \mid \boldsymbol{x}^{[k]}) &= \sum_{M \supseteq D} p_\theta(\tilde{\boldsymbol{x}}_M^{[k]} \mid \boldsymbol{x}_{\setminus M}^{[k]}) \rho(\boldsymbol{x}_{\setminus M}^{[k]} \mid \boldsymbol{x}^{[k]}, m_k) \\ &= C \sum_{M \supseteq D} p_\theta(\tilde{\boldsymbol{x}}_M^{[k]} \mid \boldsymbol{x}_{\setminus M}^{[k]}). \end{aligned} \tag{22}$$

Under the assumption that, given the unmasked tokens, the denoiser unmasks all masked tokens jointly from the exact conditional distribution under the same prior. By the standard definition of conditional probability, the joint probability of the proposed sequence under the pretrained model factorizes into the masked and unmasked components: $p_\theta(\tilde{\boldsymbol{x}}^{[k]}) = p_\theta(\tilde{\boldsymbol{x}}_M^{[k]} \mid \tilde{\boldsymbol{x}}_{\setminus M}^{[k]}) p_\theta(\tilde{\boldsymbol{x}}_{\setminus M}^{[k]})$. Rearranging this expression yields $p_\theta(\tilde{\boldsymbol{x}}_M^{[k]} \mid \tilde{\boldsymbol{x}}_{\setminus M}^{[k]}) = \frac{p_\theta(\tilde{\boldsymbol{x}}^{[k]})}{p_\theta(\tilde{\boldsymbol{x}}_{\setminus M}^{[k]})}$. Substituting this ratio into the forward proposal, and applying the established equality $\boldsymbol{x}_{\setminus M}^{[k]} = \tilde{\boldsymbol{x}}_{\setminus M}^{[k]}$, we obtain:

$$\begin{aligned} g_k(\tilde{\boldsymbol{x}}^{[k]} \mid \boldsymbol{x}^{[k]}) &= C \sum_{M \supseteq D} \frac{p_\theta(\tilde{\boldsymbol{x}}^{[k]})}{p_\theta(\boldsymbol{x}_{\setminus M}^{[k]})} \\ &= C\, p_\theta(\tilde{\boldsymbol{x}}^{[k]}) \sum_{M \supseteq D} \frac{1}{p_\theta(\boldsymbol{x}_{\setminus M}^{[k]})} \equiv C\, p_\theta(\tilde{\boldsymbol{x}}^{[k]})\, C', \end{aligned} \tag{23}$$

where $C' = \sum_{M \supseteq D} [p_\theta(\boldsymbol{x}_{\setminus M}^{[k]})]^{-1}$ is constant across all valid masks.

Applying the exact same logic and derivation to the reverse proposal kernel yields a symmetric result:

$$\begin{aligned} g_k(\boldsymbol{x}^{[k]} \mid \tilde{\boldsymbol{x}}^{[k]}) &= C \sum_{M \supseteq D} \frac{p_\theta(\boldsymbol{x}^{[k]})}{p_\theta(\tilde{\boldsymbol{x}}_{\setminus M}^{[k]})} \\ &= C\, p_\theta(\boldsymbol{x}^{[k]}) \sum_{M \supseteq D} \frac{1}{p_\theta(\boldsymbol{x}_{\setminus M}^{[k]})} = C\, p_\theta(\boldsymbol{x}^{[k]})\, C'. \end{aligned} \tag{24}$$

We now substitute these symmetric proposal functions into the model-ratio and proposal-ratio term of the exact Metropolis-Hastings acceptance probability:

$$\frac{p_\theta(\tilde{\boldsymbol{x}}^{[k]})\, g_k(\boldsymbol{x}^{[k]} \mid \tilde{\boldsymbol{x}}^{[k]})}{p_\theta(\boldsymbol{x}^{[k]})\, g_k(\tilde{\boldsymbol{x}}^{[k]} \mid \boldsymbol{x}^{[k]})} = \frac{p_\theta(\tilde{\boldsymbol{x}}^{[k]}) \big[C\, p_\theta(\boldsymbol{x}^{[k]})\, C'\big]}{p_\theta(\boldsymbol{x}^{[k]}) \big[C\, p_\theta(\tilde{\boldsymbol{x}}^{[k]})\, C'\big]} = 1. \tag{25}$$

Because the model priors and the forward/reverse proposal probabilities perfectly cancel one another, the exact acceptance probability simplifies purely to the energy difference defined by the reward-tilted target distribution:

$$A_k^{\mathrm{MH}}(\boldsymbol{x}^{[k]}, \tilde{\boldsymbol{x}}^{[k]}) = \min\left\{1, \exp\left(\beta_k\left[R(\tilde{\boldsymbol{x}}^{[k]}) - R(\boldsymbol{x}^{[k]})\right]\right)\right\}. \tag{26}$$

This concludes the proof. □

B Additional related works and comparisons

Remasking-based masked diffusion models. Masked diffusion models generate discrete data by repeatedly masking and denoising tokens, which makes partial corruption a native operation rather than an external perturbation. Large language diffusion models and remasking-based scaling methods use this mechanism to improve non-autoregressive generation, selective refinement, and controllability (Nie et al., 2026; Wang et al., 2026a; Schiff et al., 2026). These works focus on local correction and sample quality under the pre-trained model, whereas TEMPERS uses remasking as a temperature-coupled proposal scale for global exploration and local reward-driven refinement.

Steering masked and discrete diffusion models. Several recent works study reward or constraint steering for discrete diffusion models. Simple guidance mechanisms and ILRR adapt classifier or reward guidance to discrete diffusion and masked language diffusion models (Schiff et al., 2025; Avrahami & Nachmani, 2026), while DDPP frames steering for masked discrete diffusion as a posterior prediction task (Rector-Brooks et al., 2025). DRAKES fine-tunes discrete diffusion models toward biological reward functions (Wang et al., 2025), and reward-guided iterative refinement uses test-time correction for protein and DNA design (Uehara et al., 2025). Discrete Feynman-Kac Correctors and Contrastive Distribution Matching further develop SMC-style reward-tilted sampling for discrete diffusion, either through correction weights or learned twist functions (Hasan et al., 2026; Kim et al., 2026). TEMPERS is complementary to these approaches: instead of denoising from pure noise, we generate the complete sequence and perform reversible reward-aware exploration directly in the complete sequence space.

Parallel tempering in generative sampling. Parallel tempering is a classical tool for sampling from multimodal distributions (Swendsen & Wang, 1986; Earl & Deem, 2005). Recent generative-model work has begun to import this idea into diffusion and flow sampling. Source parallel tempering performs test-time guidance in the source distribution of flow-based generators (Wang et al., 2026b). PATHS uses a PT ladder to produce better initial replicas for reward-aligned SMC (Oh et al., 2026), and conditional diffusion sampling combines PT-based global exploration with diffusion-based transport (Castro-Macías et al., 2026). These works support the premise that tempered exchange is an effective response to multimodal reward landscapes.

CREPE uses replica exchange for diffusion control (He et al., 2026b). It places replicas along diffusion time and derives communication moves through path Radon-Nikodym ratios. In its reward-tilting formulation, a reward defined at clean time must be continued to intermediate diffusion states through a user-specified r_t. TEMPERS differs in where the Markov chain lives. It operates directly over complete discrete samples using MDM remask-and-denoise proposals, and couples the inverse-temperature ladder to a masking-fraction ladder so that reward strength and discrete proposal scale are coordinated. Furthermore, because absorbing diffusion enables complete candidate reconstruction in single unmasking mini-steps rather than multi-step denoising trajectories, TEMPERS achieves tempered exploration within compute budgets comparable to standard generation.

C Experimental details

C.1 Analytic examples

Dataset. We simulate a DNA-like dataset: length-18 sequences over the DNA alphabet $\{A, T, C, G\}$, with an analytic prior defined by four neighborhoods $B_1, \ldots, B_4$ in Hamming space. The four centers lie on a Hamming path: adjacent centers differ at six positions, and the first and last differ at all 18 positions. Figure 4 illustrates this landscape with small neighborhoods. For the

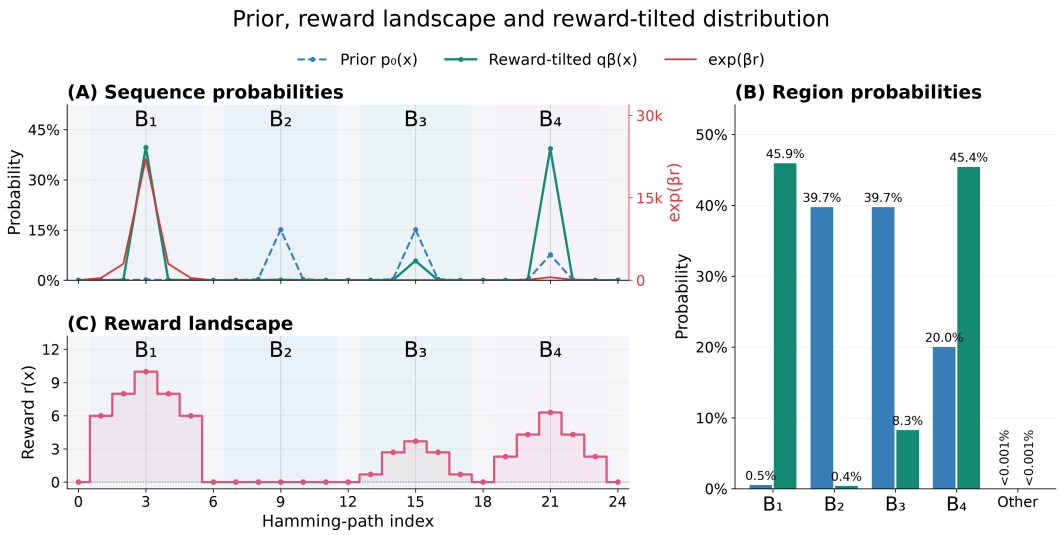


Figure 4: (A) Prior and reward-tilted probabilities along the Hamming path. (B) Region probabilities over the full support. (C) Reward landscape along the same path.

quantitative comparison in Figure 1, we use a wider-support version: the four prior components have masses approximately $5\%, 35\%, 35\%, 20\%$, Hamming radii $3, 3, 4, 4$, and per-sequence weights that decrease by a factor of 0.02 per additional mismatch. The remaining 5% of the prior mass is assigned to sequences on the displayed path that lie outside those neighborhoods. This prior has 588,034 sequences in its support; the five evaluation regions are $B_1, \ldots, B_4$ and Other. Figure 4 demonstrates the prior distribution.

Pre-trained model. A four-layer MDM with hidden dimension 128 and four attention heads is trained on samples from this analytic prior. We use the checkpoint selected at training step 15,500 from a 16,000-step run. Generating a fresh complete sequence takes 20 denoiser evaluations.

Reward function. The reward is a piecewise function of Hamming distance to the four centers: B_1 and B_4 receive high scores, B_3 receives intermediate scores, and B_2 receives a low score. For the quantitative comparison, the scores at distances $0, 1, 2$ are $(7.673, 5.673, 3.673)$ for B_1, $(3.801, 2.801, 0.801)$ for B_3, and $(6.3, 4.3, 2.3)$ for B_4; B_2 receives 1.033 through distance three, and supported sequences in Other receive 2.979. Unlisted scores within B_1, B_3, B_4 are zero. With $\beta = 1$, we enumerate the prior support to compute the exact target $q_\beta(\boldsymbol{x}) \propto p_0(\boldsymbol{x}) \exp\{\beta R(\boldsymbol{x})\}$. Its masses in $(B_1, B_2, B_3, B_4, \text{Other})$ are approximately $(45, 1, 8, 45, 1)\%$.

Methods and NFE. One network function evaluation (NFE) means one denoiser evaluation for one sequence, including calls made during initialization and warm-up. Let $T = 20$ be the number of denoising steps for a fresh sample, n the number of unguided sampling runs or Best-of-K selection groups, B the number of candidates per group, P the number of FK/SMC particles, K the number of replicas, and S the number of update sweeps. Unguided sampling, Best-of-K, and FK/SMC cost nT, nBT, and PT, respectively. TEMPERS costs $K(T + S)$ with one denoising mini-step per proposal; exchanges require no additional denoiser evaluations. CREPE costs $KT + \sum_{s=1}^{S} c_s$, where c_s counts the denoiser evaluations in exchange sweep s. Table 4 substitutes the settings used for the two budgets. The eight warm-up sweeps of TEMPERS and CREPE are included in S.

Table 4: Denoiser NFE per independent system in the analytic comparison. Each budget includes initialization and warm-up.

Method	NFE formula	200-NFE budget	400-NFE budget
Unguided MDM	nT	$10 \cdot 20 = 200$	$20 \cdot 20 = 400$
Best-of-K	nBT	$5 \cdot 2 \cdot 20 = 200$	$5 \cdot 4 \cdot 20 = 400$
FK/SMC	PT	$10 \cdot 20 = 200$	$20 \cdot 20 = 400$
TEMPERS	$K(T+S)$	$5(20+20) = 200$	$10(20+20) = 400$
CREPE	$KT + \sum_s c_s$	$100 + 98 = 198$	$200 + 196 = 396$

Best-of-K keeps the highest-reward candidate from each group. FK/SMC uses reward differences for particle reweighting with scale $\alpha = 1$. TEMPERS uniformly chooses positions to remask, fills them in one mini-step, and attempts exchanges between neighboring replicas after every sweep; its output is the cold chain. CREPE uses independently initialized diffusion levels, a linear intermediate-reward schedule, and alternating neighboring exchanges. Both replica methods use five levels at the 200-NFE budget and ten at the 400-NFE budget, as in Figure 1.

Evaluation. For the region comparison, we run eight seeds with 256 independent systems per seed and retain five complete outputs from each system. We compare the resulting five-region output histogram $\hat{p}$ with the exact target region masses q; the KL displayed in Figure 1 is $D_{\mathrm{KL}}(q\|\hat{p}) = \sum_r q(r) \log[q(r)/\hat{p}(r)]$. Error bars show the standard deviation across seed means. Separately, Figure 2a follows 2,048 independent systems over 40 sweeps: it shows mean output reward, reward change from initialization for selected systems, and the rewards of complete exchange proposals before acceptance, including rejected proposals.

C.2 Protein sequence design

Dataset. We use the 12 test protein backbones from the Megascale split released with DRAKES (Wang et al., 2025; Tsuboyama et al., 2023). Each backbone conditions sequence generation. For each method and seed, we generate 128 final sequences per target in eight batches of 16, giving 1,536 sequences per seed and 4,608 across seeds $\{42, 1004, 1126\}$. The candidates, particles, or replicas used internally are not counted as additional outputs.

Pre-trained model. We use the frozen inverse-folding MDM released by DRAKES, which uses the ProteinMPNN architecture and was trained on 23,349 PDB structures and their sequences (Wang et al., 2025; Dauparas et al., 2022). It has three encoder and three decoder layers, hidden dimension 128, and 30 structural neighbors. Sampling starts from a fully masked sequence with a linear schedule on 50 time points from 0.01 to 1; this requires $T = 49$ denoiser evaluations. No generator weights are updated.

Reward function. The reward predicts stability relative to the wild-type sequence, following DRAKES's convention that larger DDG is better. We use the released guidance oracle for sampling and a separate evaluation oracle for the DDG values in Table 2. Both take a sequence and its target backbone as input. The guidance oracle was trained on the Megascale training split; the evaluation oracle was trained on the full curated dataset, including variants of the test backbones (Wang et al., 2025). The reported DDG is therefore a model prediction.

Methods and NFE. TEMPERS independently initializes $K = 10$ replicas at sampling temperature 0.3, which is also used for proposal refills. It runs $S = 50$ sweeps with uniform remasking and one denoising mini-step per proposal. The inverse temperatures increase linearly from 50 to 3333, while masking fractions decrease linearly from 0.2 to 0.05. Every sweep attempts exchanges sequentially between all adjacent pairs, from the hottest end to the coldest, and the final cold-chain sequence is returned.

CREPE uses ten independently initialized diffusion levels, base sampling temperature 0.5, proposal temperature 0.25, and reward scale 3333. Its intermediate reward is weighted by $\gamma(t) = (1-t)^4$, with noise level $t = 0$ denoting a complete sequence and $t = 1$ a fully masked sequence. Masked residues are filled by the denoiser's most likely predictions for reward evaluation. The levels use

noise-grid indices $(0, 5, 11, 16, 22, 27, 33, 38, 44, 49)$ divided by 49. We perform 50 alternating odd/even exchange updates (25 of each) and return the final complete sequence at level zero.

Evaluation. For each seed, DDG is the median evaluation-oracle score over all 1,536 outputs. To measure structural fidelity, we fold each sequence with ESMFold (Lin et al., 2023), apply the DRAKES PyRosetta side-chain repacking and FastRelax procedure to the predicted and target structures, and compute their backbone RMSD. We report the median scRMSD, the percentages with DDG> 0 and scRMSD< 2, and the success rate satisfying both conditions. Each table entry is the mean and population standard deviation of the three seed-level statistics.

Diversity is the mean pairwise Hamming distance among the 128 final sequences for the same backbone, divided by the number of designed residues and expressed as a percentage. We exclude self-pairs, retain duplicate sequences, and average equally over the 12 backbones before summarizing across seeds. All final sequences are included, regardless of reward or structural validity.

More results. Figure 2b follows 1,536 independent systems for seed 42 over 50 updates with the TEMPERS and CREPE settings above. It shows median output reward, reward change from initialization for selected systems, and rewards of complete exchange proposals before acceptance, including rejected proposals. These plots use the evaluation oracle; sampling decisions continue to use the guidance oracle. Figure 5 separately varies the temperature and masking ladders at a fixed denoiser cost.

C.3 DNA SEQUENCE DESIGN

Dataset. We follow the DRAKES regulatory-enhancer protocol (Wang et al., 2025) and design 200 bp sequences over $\{A, C, G, T\}$ that drive expression in the HepG2 cell line. Every method generates 640 sequences per seed in ten batches of 64, giving 1,920 sequences across seeds $\{42, 1004, 1126\}$. The candidates, particles, or replicas used internally are not counted as additional outputs.

Pre-trained model. We use the frozen enhancer MDM released by DRAKES, a dilated-convolution denoiser with 20 layers of width 128 and kernel size 9, trained on the 735,156 measured enhancers of the accompanying dataset. Sampling starts from a fully masked sequence and follows the released DDPM sampler with a log-linear noise schedule on 128 steps, which costs $T = 257$ denoiser evaluations: two per step plus one final argmax readout. No generator weights are updated.

Reward function. The reward is the predicted HepG2 enhancer activity of a complete sequence. DRAKES releases two Enformer-based oracles of the same architecture trained on disjoint chromosome halves of the dataset; we steer with one and report Pred-Activity with the other, which never enters sampling. The reported activity is therefore a model prediction.

Methods and NFE. *TEMPERS* independently initializes K=10 replicas and runs S=100 sweeps with uniform remasking and one denoising mini-step per proposal. The inverse temperatures increase linearly from 0.05 to 3 and the masking fractions decrease linearly from 0.8 to 0.05, that is, from 160 to 10 of the 200 positions. Every sweep attempts exchanges sequentially between all adjacent pairs, and the final cold-chain sequence is returned.

CREPE uses ten independently initialized diffusion levels with reward scale β=3, proposal temperature 1, and an intermediate reward weighted by $\gamma(s) = (1 - s)^4$, where s is the normalized masking probability, s=0 for a complete sequence and s=1 for a fully masked one. The levels sit at steps $(0, 14, 28, 43, 57, 71, 85, 100, 114, 128)$ of the same 128-step grid, and we perform 300 alternating odd/even exchange updates and return the complete sequence at level zero.

Evaluation. For each seed, Pred-Activity is the median evaluation-oracle score over the 640 outputs. ATAC-Acc is the percentage of outputs that the binary chromatin-accessibility classifier released with DRAKES calls accessible in HepG2 (ENCODE Project Consortium et al., 2012). 3-mer Corr is the Pearson correlation between the 3-mer counts of the outputs and those of the 736 dataset sequences above the 99.9th percentile of HepG2 activity, rescaled to the same number of sequences. Diversity is the mean pairwise Hamming distance over all pairs of the 640 outputs of a seed, divided by 200

and expressed as a percentage; we exclude self-pairs and retain duplicate sequences. Each table entry is the mean and population standard deviation of the three seed-level statistics.

More results. Scanning the same reference set with the JASPAR2022 CORE database gives a motif-count correlation of 0.962 ± 0.002 for TEMPERS, 0.953 ± 0.002 for CREPE, and 0.944 ± 0.003 for FK-Steering, in the same order as the 3-mer column of Table 1. All 640 outputs of every run are distinct, so the diversity column is not inflated or deflated by repeated sequences.

C.4 TEXT SENTIMENT CONTROL

Dataset. We steer LLaDA-8B-Instruct (Nie et al., 2026), a pretrained masked diffusion language model, toward positive and negative sentiment on 60 English IFEval prompts (Zhou et al., 2023). Each IFEval prompt carries one to three verifiable instructions, such as a word limit, a forbidden keyword, or an all-lowercase constraint, which are checked by rules rather than by a model. Each prompt is steered toward positive and toward negative sentiment, which gives 120 prompt–direction pairs. The prompt is wrapped in the LLaDA-8B-Instruct chat template, and every method generates a response of 128 tokens.

Pre-trained model. All methods use the frozen LLaDA-8B-Instruct checkpoint without classifier-free guidance. Generation follows LLaDA's low-confidence remasking: at every step the model predicts all masked positions, a fixed number of the most confident predictions is committed, and end-of-text tokens are committed last, so that every response is a complete 128-token block after the last step. The pretrained reference decodes greedily in $T{=}50$ steps.

Reward function. The reward is the DistilBERT sentiment classifier of Section 4.2.3, a DistilBERT-base model fine-tuned on SST-2.[2] For a decoded response $\boldsymbol{x}$, let $p \in [\frac{1}{2}, 1]$ be the classifier's confidence in its predicted label. The reward is $r(\boldsymbol{x}) = +p$ when the label is POSITIVE and $r(\boldsymbol{x}) = -p$ when it is NEGATIVE; the negative direction uses $-r(\boldsymbol{x})$, so higher is better in both directions.

Methods and NFE. *CREPE* (He et al., 2026b) uses 10 noise levels $t_m = m/9$ with reward strength $\beta(1 - t_m)$ at level m and $\beta{=}8$ at the clean level. Every level is initialized by its own T-step denoising path, after which $S{=}64$ exchange sweeps attempt every other adjacent pair of levels in turn, so a sweep costs $c_s{=}9$ or 8 denoiser evaluations. A pair proposal forward-masks the cleaner state and denoises the noisier one with tokens sampled at temperature 0.7, and is accepted by the RNE ratio of He et al. (2026b), the change of the two tilted potentials plus the kernel correction that a proposal temperature below one requires.

TEMPERS runs $K{=}10$ replicas on a linear inverse-temperature ladder from $\beta_1{=}0.1$ to $\beta_{10}{=}8$ and a linear masking ladder from $m_1{=}0.4$ on the hottest replica to $m_{10}{=}0.05$ on the coldest, that is, from 51 to 6 remasked tokens. Each replica is initialized by T low-confidence denoising steps with tokens sampled at temperature 0.7, keeping the greedy end-of-text decisions, and then takes $S{=}100$ MCMC sweeps. Each sweep remasks a uniformly random subset of every replica, refills it in one denoising pass at temperature 0.7, applies the acceptance rule $\min\{1, \exp(\beta_k \Delta R_k)\}$ of Section 3, and then proposes exchanges between neighbouring replicas, alternating the even and odd pairs between sweeps.

Evaluation. **Sentiment** is the target-direction reward r of the returned response, so it is scored by the steering classifier itself. **Strict** and **Loose** are IFEval prompt-level accuracies computed with the official checker: a pair counts as correct only if the response satisfies every instruction of its prompt, and the loose variant also accepts the response after removing markdown asterisks and after dropping its first line, its last line, or both. Cells report means over the evaluated prompt–direction pairs, with standard errors computed as $s/\sqrt{n}$, where s is the sample standard deviation and n is the number of pairs included in the corresponding statistic. The PPL column of Table 3 is scored by LLaDA-8B-Instruct itself, with the prompt in context, using the conditional likelihood bound of Nie et al. (2026).

[2]Sentiment reward: DistilBERT `https://huggingface.co/okho0653/distilbert-base-uncased-finetuned-sst-2-english-zero-shot-sentiment-model`

More results. Table 5 shows the greedy pretrained response and the two steered responses of TEMPERS for three of the 60 prompts, with the classifier score s of each response ($s > 0$ is positive sentiment) and its strict IFEval result. Because the initialization does not use the reward, both directions start from the same ten sampled responses, and the two steered answers of the first prompt differ in three words, which is what remasking a few tokens of a complete sentence and refilling them under the reward produces. For the other two prompts the greedy response degenerates into repetition and fails its instruction, while both steered responses satisfy the instruction and carry the requested sentiment.

Table 5: Pretrained and TEMPERS responses for three IFEval prompts. The pretrained response is greedy and therefore identical for both directions. s is the sentiment classifier's score of the response, positive for positive sentiment; Strict is the IFEval strict check. Texts marked [. . .] are shortened.

Response	Text	s	Strict
Prompt: Is praying for someone's health a good idea? Your answer must be in all capital letters and in English.			
Pretrained	YES, PRAYING FOR SOMEONE'S HEALTH IT IS A GOOD IDEA.	$+0.999$	pass
TEMPERS (pos.)	YES, PRAYING FOR SOMEONE'S HEALTH IS A GOOD IDEA.	$+0.990$	pass
TEMPERS (neg.)	YES, PRAYING FOR SOMEBODY'S HEALTH IS A USELESS ACT.	-1.000	pass
Prompt: Write a joke about morphology that's professional and includes the word "cat" at least once, and the word "knock" at least twice. Wrap your whole response with double quotation marks.			
Pretrained	"Why did the morphology go to the cat? Because it heard the knock on the door and wanted to get the morphology on the morphology. But,, it,, it,, it,, [. . .]"	-0.999	fail
TEMPERS (pos.)	"Why did the cat play knock-knock? Because it wanted to knock down the morphological structures with a 'gentle, gentle'!"	$+0.968$	pass
TEMPERS (neg.)	"Why did the cat decide not to knock on the door? Because it knew it would just be a knock on the morphological!"	-0.995	pass
Prompt: Create a dialogue between two people who are both trying to avoid using the letter t. But somehow they ended up using a lot of t in their dialogue. Break the dialogue into two scenes, separated by 6 asterisk symbols: ******. The letter t must appear at least 30 times in the dialogue you write.			
Pretrained	Scene 1: Person: Hey, have you seen the the movie?? Person: Yes, I have. It's the great. [. . .]	$+1.000$	fail
TEMPERS (pos.)	Scene 1: A: Hey, have you seen that latest movie? B: Yes, it's great. Have you seen it? A: I saw it, and it's fantastic. B: I saw it, and it's just amazing. [. . .]	$+1.000$	pass
TEMPERS (neg.)	Scene 1: Person: Hey, have you seen the new movie? Person: Yeah, I saw it last night. But it was terrible. Person: Really? I didn't think it was terrible. [. . .]	-0.999	pass

C.5 More Ablation study

Proposal denoising steps The remark in Section 3 completes each proposal in a single denoising mini-step. Table 6 varies only the number of mini-steps T_{prop} used to refill a proposal on inverse protein folding, keeping the $K{=}10$ replicas, the $S{=}50$ sweeps, and both ladders fixed, so that each returned sequence costs $K(T + S\,T_{\text{prop}})$ denoiser evaluations. Raising T_{prop} from 1 to 50 multiplies that cost by 26 without improving the reward: DDG is highest at $T_{\text{prop}}{=}1$ and the success rate moves within 1.8 points with no monotone trend, while scRMSD changes by at most 0.018 Å. Longer refills give the denoiser more freedom to revert the remasked positions to a high-likelihood completion, which weakens the reward signal that the acceptance test in Eq. (16) acts on, so we use one mini-step throughout the paper. This comparison uses seed 1126 and the untuned uniform configuration, so its absolute values sit above the tuned row of Table 2; only the trend across T_{prop} is being read.

Table 6: Proposal denoising steps on inverse protein folding. Only T_{prop}, the number of denoising mini-steps per proposal, varies; NFE is the denoiser cost per returned sequence.

T_{prop}	Pred-ddG ↑	%(ddG>0) ↑	scRMSD ↓	%(scRMSD<2) ↑	Success Rate ↑	NFE
1	1.2488	81.71	0.8948	87.24	70.05	990
5	1.2375	79.36	0.8771	91.08	71.22	2,990
10	1.1288	77.41	0.8873	89.71	69.53	5,490
50	1.1664	78.78	0.8776	90.56	71.29	25,490

Temperature ladder sensitivity The core of TEMPERS is to perform parallel tempering on the reward temperature of the complete sequence, using remask-and-denoise as the proposal. Because the complete-sequence predictor of MDM is time-independent (Sec 2), the remasking ratio can be set independently for each chain. Fig 5 shows that the reward is primarily driven by the β ladder: the

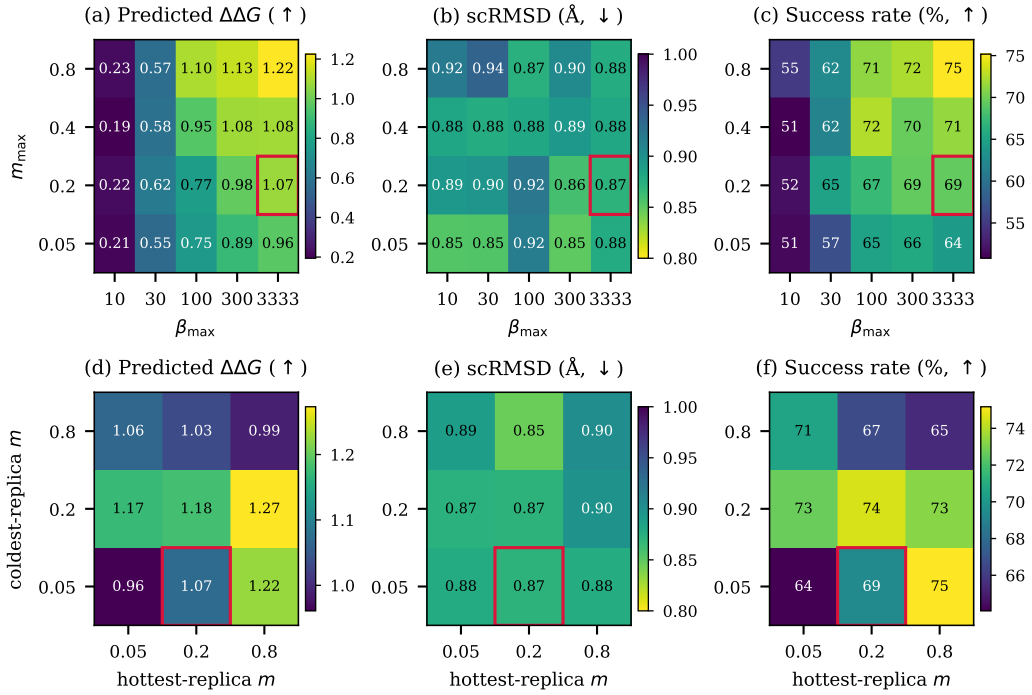


Figure 5: Ladder sensitivity of TEMPERS on inverse protein folding. Each cell changes only the two ladders of the tuned configuration shown in Table 2 (boxed), so all cells make the same number of denoiser calls. Top row: the inverse-temperature ladder is rescaled as a whole ($\beta_{\min} = 0.015\,\beta_{\max}$), and the masking ladder runs linearly from $m_{\max}$ on the hottest chain to $m_{\min} = 0.05$ on the coldest. Bottom row: at the tuned $\beta \in [50, 3333]$, the masking fractions of the hottest and the coldest chain are set independently, with the ladder's row for the coldest chain, $m = 0.05$, repeats the $\beta_{\max} = 3333$ column of the top row. (a, d) Median DDG. (b, e) Median scRMSD, on a fixed 0.80–1.00 color scale. (c, f) Success rate (DDG > 0 and scRMSD < 2).

coldest chain (i.e., the output) requires a moderate step size; too small (0.05) is difficult to adjust, and too large (0.8) will almost always be rejected at high β. When the step size of the cold end is small, the hot end can compensate by performing a large rewrite through exchange. Increasing the inverse temperature β_k significantly improves performance, after which the gains saturate. When reward gains saturate, the effect of the remasking ratio m_k is more apparent: a larger rewrite at the hot end helps in the top-row setting, but too much rewrite at the cold end actually degrades performance. scRMSD varies little between these settings, so the main differences lie in the reward and success rate.