VALUE-INFORMED SCHEDULE TRIMMING FOR ACCELERATED SEQUENTIAL MONTE CARLO GUIDANCE IN DISCRETE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-guided discrete diffusion enables test-time alignment with downstream objectives, but applying Sequential Monte Carlo (SMC) guidance at every denoising step is computationally expensive. Active guidance steps require additional reward evaluations and, for gradient-based proposals, value-gradient computations. Sparse guidance reduces this cost by guiding only a subset of timesteps, but commonly used schedules are fixed heuristics whose effectiveness can vary substantially across tasks. We propose VISTA, a training-free method that automatically selects guided timesteps under a fixed budget. VISTA formulates sparse guidance as approximation of the ideal full-guidance probability path and derives an estimable upper bound on ℓ_1 path deviation. Using value statistics collected from a small number of full-guidance warmup generations, VISTA optimizes the resulting empirical surrogate exactly via dynamic programming. The warmup generations can be retained as model outputs, and after schedule selection VISTA has the same per-generation guidance complexity as other sparse schedules with the same budget. Across text, biological sequence, and image generation tasks, VISTA discovers task-dependent schedules and outperforms hand-designed sparse schedules in most tested settings, while approaching or exceeding full-guidance performance on several task metrics. In toxic-text generation, for example, VISTA attains a 0.99 unique-toxic ratio compared with 1.00 for full guidance while reducing post-selection per-generation latency from 204.23 s to 38.16 s, a 5.35$\times$ speedup.

1 INTRODUCTION

Discrete diffusion models (Austin et al., 2021) have emerged as a powerful framework for generating structured objects such as text, biological sequences, and discrete images. In these domains, practitioners often prioritize not only the likelihood under a pretrained model but also alignment with external objectives, such as safety in language, biological activity in DNA, or constraint satisfaction. Consequently, a primary goal is to sample from a reward-tilted distribution that steers the pretrained model toward higher expected rewards while preserving its generative capabilities.

Several reward-tilted guidance techniques for discrete domains have recently been introduced (Nisonoff et al., 2024; Schiff et al., 2024). However, these independent sampling methods—which generate samples along separate trajectories—often rely on derivative-based approximations that can introduce significant error and bias (Lee et al., 2025a). To address this limitation, Sequential Monte Carlo (SMC) methods have been proposed. SMC begins by generating multiple candidate particles using a proposal policy. To align the particles with the reward-tilted distribution, it employs a resampling mechanism: particles with high value functions are duplicated, while those with low values are discarded (see Figure 1-(b)). Under the asymptotic conditions used in the SMC analysis, this framework recovers the exact reward-tilted distribution at guided timesteps (Lee et al., 2025a; Ou et al., 2025; Naesseth et al., 2019; Feng et al., 2024; Kim et al., 2025b; Luo et al., 2026).

Despite their theoretical elegance, the computational cost of SMC guidance remains a bottleneck in practical applications. Each guided denoising step requires not only a pretrained diffusion transition but also an evaluation by an external reward model; moreover, gradient-based proposals often necessitate additional backward passes. Applying this correction at every timestep t can dominate

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

the total wall-clock time. Consequently, sparse guidance—where only a subset of $T' < T$ timesteps is guided—provides a natural way to improve the efficiency of inference-time alignment, where T denotes the total number of diffusion steps. This approach can reduce the total reward and gradient computation to a fraction of T'/T (see Table 2 for a detailed breakdown).

However, a central question remains under-explored: which steps should be guided? Existing sparse guidance schedules are typically heuristic-based and fixed, for example by restricting guidance to a prescribed interval (Kynkäänniemi et al., 2024; Hoogeboom et al., 2024; Yu et al., 2024) or applying it at uniformly spaced timesteps (Singhal et al., 2025; Hemmat et al., 2024; Kim et al., 2025a). Such fixed schedules can be suboptimal because the effective timing of guidance depends on the task and reward landscape. As we show empirically, some tasks favor guidance concentrated in the early denoising stages, whereas others benefit from combining early guidance with corrections distributed throughout the reverse process. Ultimately, a schedule that is effective for one reward landscape can be suboptimal for another.

To mitigate this issue, we propose VISTA-SMC (Value-Informed Schedule Trimming for Accelerated Sequential Monte Carlo), a training-free, plug-and-play procedure for identifying a budgeted schedule of guided steps. By minimizing a tractable surrogate for the discrepancy between the full-guidance trajectory and the reduced-guidance path, VISTA characterizes the guidance dynamics of a given task and identifies a highly effective schedule tailored to its specific reward landscape.

Specifically, VISTA addresses sparse SMC guidance as a full-path approximation problem: it chooses the guided-step set (i.e., $\mathcal{G}$) so that the reduced-guidance path stays close to the ideal full-guidance path (i.e., $\{\pi_t^*\}_{t=0}^T$). Directly optimizing this gap is infeasible because the ground-truth full-guidance marginals are not available as explicit distributions over the discrete state space. Our main observation is that the gap admits a tractable upper bound expressed through value expectations along the full-guidance path. These expectations can be estimated from a small number of warmup full-guidance runs; the resulting empirical budgeted objective is then solved exactly by dynamic programming. The selected schedule is reused for all later generations with the same model, reward, budget, and proposal family, so the warmup cost is amortized and the per-generation comparison to sparse heuristics is fair.

Our contributions are ordered by conceptual importance:

1. **Automatic budgeted guidance scheduling.** We introduce a training-free scheduler that automatically selects a task-specific subset of SMC guidance timesteps under a fixed budget, using full-guidance value statistics instead of a predefined temporal pattern.
2. **Full-guidance gap formulation.** We view sparse SMC guidance as approximating the ideal full-guidance probability path, not only the final marginal. This is crucial: final-only correction may be asymptotically valid yet can lead to diversity collapse with finite particles.
3. **Tractable gap minimization without ground-truth marginals.** To our knowledge, VISTA is the first sparse-SMC scheduler to target the gap to full-guidance marginals without accessing those intractable marginals. We prove an estimable upper bound on ℓ_1 path deviation and optimize the resulting surrogate under a fixed budget.
4. **Practical schedule construction.** VISTA uses a few initial full-guidance generations to estimate a task-specific schedule. These generations are retained as generated outputs rather than discarded preprocessing. Afterward, VISTA has the same per-generation complexity as heuristic sparse schedules with the same budget, although heuristics can operate sparsely from the first generation.
5. **Task-specific empirical schedules.** On Binary MNIST, toxic text, and DNA sequence generation, VISTA finds reward-dependent schedules and outperforms hand-designed sparse baselines in most tested settings at the same budget. It also yields large speedups over full guidance, including a $5.35\times$ post-selection speedup in the toxic-text SMC-grad experiment at $T' = 15$, while achieving a unique-toxic ratio of 0.99 compared with 1.00 under full guidance.

2 Related Work

Discrete and Masked Diffusion. Diffusion models define a forward process that transforms a data distribution into a tractable prior and learn reverse transitions for generation. While continuous

diffusion models typically use Gaussian perturbations and score-based reverse dynamics (Ho et al., 2020; Song et al., 2020), discrete diffusion models define Markov transitions over categorical or masked states (Austin et al., 2021; Sahoo et al., 2024). Masked diffusion language models instantiate this idea by using a special [MASK] token as the high-noise state and iteratively unmasking or denoising tokens. Our work focuses on inference-time reward guidance for such reverse processes, with a particular emphasis on addressing the challenges of discrete state spaces.

Sequential Monte Carlo Methods. SMC samplers propagate particles through a proposal policy and use resampling to correct bias, effectively aligning the particles with the target reward-tilted distribution (Del Moral et al., 2006; Naesseth et al., 2019; Chopin et al., 2020). Recent work has adapted SMC to diffusion inference for tasks such as continuous and discrete diffusion steering, test-time alignment, inverse problems, and tree-structured search (Ou et al., 2025; Skreta et al., 2025; Hasan et al., 2026; Achituve et al., 2025; Wu et al., 2025; Jain et al., 2025; Guo et al., 2025; Chanchu et al.). While these methods primarily focus on *how* particles are weighted or proposed, VISTA is complementary: it automatically identifies the most critical timesteps for guidance, determining *when* these expensive corrections should be executed to maximize overall efficiency.

Inference-time Acceleration of Diffusion Guidance. Prior work has reduced the cost of diffusion guidance using hand-designed temporal schedules, including restricting guidance to a prescribed interval (Kynkäänniemi et al., 2024; Hoogeboom et al., 2024; Yu et al., 2024) or uniformly spaced timesteps (Singhal et al., 2025; Hemmat et al., 2024; Kim et al., 2025a). Other approaches reduce guidance overhead through adaptive rules, caching, or multirate integration (Castillo et al., 2025; Lv et al., 2024; Lee et al., 2025b). These methods demonstrate that guidance need not be applied at every denoising step, but their schedules are either manually specified or developed primarily for continuous-image guidance and network-evaluation reuse. In contrast, VISTA is specifically formulated for reward-guided SMC in discrete state spaces and provides a theoretical bound on the probability-path deviation. Table 1 summarizes these key distinctions.

Table 1: Comparison of VISTA with existing guidance acceleration methods. Discrete: Designed for discrete state spaces; Reward: Handles external reward functions; Theoretical Guarantee: Provides a rigorous theoretical justification for the acceleration schedule.

Method	Discrete	Reward	Theoretical Guarantee
Adaptive Guidance (Castillo et al., 2025)	×	×	×
FasterCache (Lv et al., 2024)	×	×	×
Tortoise-and-Hare (Lee et al., 2025b)	×	×	✓
Guidance Interval (Kynkäänniemi et al., 2024)	×	×	×
VISTA (Ours)	✓	✓	✓

3 PRELIMINARIES

3.1 REFERENCE REVERSE PROCESS

Let $\mathcal{X}$ be a discrete state space. A pretrained reverse diffusion model defines a reference path:

$$p_T(x_T) \prod_{t=0}^{T-1} p_{t|t+1}(x_t|x_{t+1}), \tag{1}$$

with one-step reverse kernels $p_{t|t+1}$ and marginals p_t. For $0 \le s \le t \le T$, we denote by $p_{s|t}$ the composed transition from x_t to x_s. In a masked diffusion model, x_T is typically a fully masked sequence, and x_0 is a clean sequence.

3.2 REWARD-TILTED FULL-GUIDANCE PATH

Given a reward function $r(x_0)$ for a sample x_0, our objective is to maximize the expected reward while regularizing the divergence from the pretrained model's prior. Specifically, we aim to find an optimal policy $\{\pi_t\}_{t=T}^0$ that maximizes:

$$\mathbb{E}_{\{\pi_t\}_{t=T}^{0}}\left[r(x_0) - \alpha \sum_{t=0}^{T-1} \mathrm{KL}(\pi_{t|t+1}(\cdot|x_{t+1}) \parallel p_{t|t+1}(\cdot|x_{t+1})) - \alpha \mathrm{KL}(\pi_T(\cdot) \parallel p_T(\cdot))\right], \quad (2)$$

where $\alpha > 0$ is a temperature parameter controlling the trade-off between reward maximization and prior preservation (Fan et al., 2023; Uehara et al., 2024a;b). The closed-form solution to this optimization problem is:

$$\pi^*_{t|t+1}(x_t|x_{t+1}) = \frac{V_t(x_t)}{V_{t+1}(x_{t+1})} p_{t|t+1}(x_t|x_{t+1}), \quad t \in \{0, \dots, T-1\} \quad (3)$$

$$\pi^*_t(x_t) = \frac{1}{Z} V_t(x_t) p_t(x_t), \quad (4)$$

where $Z := \mathbb{E}_{x_0 \sim p_0}[R(x_0)]$ is the partition function. Here, $R(x_0) := \exp(r(x_0)/\alpha)$ represents the exponentiated reward, and $V_t(x_t)$ is the exponentiated value function defined as:

$$V_t(x_t) := \mathbb{E}_{x_0 \sim p_{0|t}(\cdot|x_t)}\left[R(x_0)\right]. \quad (5)$$

Although the value function $v_t(x_t) := \alpha \log V_t(x_t)$ is standard in the literature (Uehara et al., 2024b; 2025), the exponentiated notation V_t simplifies our derivation.

For a detailed derivation, we refer the reader to (Uehara et al., 2024b) or our Appendix A.1.

Definition 3.1. We call π^*_t the **full-guidance path**. It is the gold-standard path that VISTA aims to approximate.

3.3 SMC Proposal Families

Algorithm 1 SMC guidance

1: $x_T^{(i)} \sim \pi^*_T(x_T) \quad (i = 1, 2, \cdots, N)$
2: **for** $t = T-1, T-2, \cdots, 0$ **do**
3: Propagation: $x_t^{(i)} \overset{\text{iid}}{\sim} q_{t|t+1}(x_t|x_{t+1}^{(i)}) \quad (i = 1, 2, \cdots, N)$
4: Resampling: Duplicate particles with high V_t and discard those with low V_t to ensure $x_t^{(i)} \sim \pi^*_t(x_t)$
5: **end for**

SMC guidance propagates particles through a proposal policy $q_{t|t+1}$ and utilizes the resampling to align the particle distribution with the target path π^*_t (Algorithm 1; see Algorithm 4 for details). Generally, the closer $q_{t|t+1}$ is to $\pi^*_{t|t+1}$, the higher the sampling efficiency. We evaluate VISTA on two proposal families established in recent discrete SMC literature (Ou et al., 2025):

- **SMC-base** uses the pretrained transition: $q_{t|t+1}(x_t|x_{t+1}) = p_{t|t+1}(x_t|x_{t+1})$.
- **SMC-grad** uses a first-order Taylor approximation of the value function to steer the proposal, yielding $q_{t|t+1}(x_t|x_{t+1}) \propto p_{t|t+1}(x_t|x_{t+1}) \exp(\frac{1}{\alpha} \nabla v_{t+1}(x_{t+1}) \cdot x_t)$.

VISTA is proposal-agnostic; it does not modify the underlying proposal family. Instead, it strategically selects which timesteps should incur the computational overhead of value estimation and resampling to maximize efficiency.

4 VISTA: Value-Informed Schedule Trimming

In this section, we introduce Reduced SMC Guidance, a framework designed to accelerate the inference of guided diffusion models by strategically skipping guidance steps.

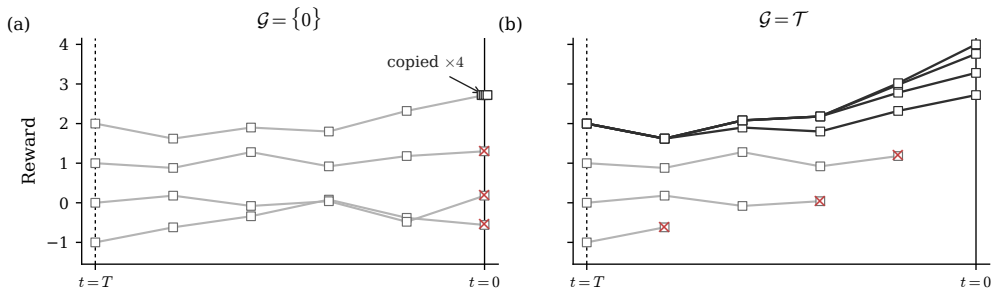


Figure 1: Impact of guidance schedules on particle distributions. A minimal schedule (e.g., $\mathcal{G} = \{0\}$) is asymptotically correct as $N \to \infty$, but with finite particles, it leads to diversity collapse through the duplication of a few terminal particles. In contrast, full guidance ($\mathcal{G} = \mathcal{T}$) maintains intermediate marginals close to the reward-tilted path, ensuring both high reward and sample diversity.

4.1 Sparse guidance as full-path approximation

Let $\mathcal{T} = \{0, 1, \cdots, T-1\}$ denote the set of reverse timesteps. A sparse schedule is a subset $\mathcal{G} \subset \mathcal{T}$ with budget $|\mathcal{G}| = T'$. Under this framework, reward guidance is applied if and only if $t \in \mathcal{G}$, as detailed in Algorithm 2.

Algorithm 2 Diffusion Guidance Reduction

$x_T^{(i)} \sim \pi_T^*(x_T) \quad (i = 1, 2, \cdots, N)$
for $t = T-1, T-2, \cdots, 0$ **do**
  Guidance is ON $\leftarrow t \in \mathcal{G}$
  if Guidance is ON **then**
    Propagation: $x_t^{(i)} \overset{\text{iid}}{\sim} q_{t|t+1}(x_t|x_{t+1}^{(i)}) \quad (i = 1, 2, \cdots, N)$
    Resampling: Adjust particles $\{x_t^{(i)}\}$ to ensure $x_t^{(i)} \sim \pi_t^*(x_t)$
  else
    Propagation: $x_t^{(i)} \overset{\text{iid}}{\sim} p_{t|t+1}(x_t|x_{t+1}^{(i)}) \quad (i = 1, 2, \cdots, N)$
  end if
end for

For any $t \in \mathcal{T}$, we define the most recent active step $\lceil t \rceil$ as:

$$\lceil t \rceil := \min\{s \in \mathcal{G} \cup \{T\} \mid s \geq t\}. \tag{6}$$

If $t \in \mathcal{G}$, then $\lceil t \rceil = t$ and the sampler performs a guided SMC correction. If $t \notin \mathcal{G}$, it propagates from $\lceil t \rceil$ to t using unguided reference transitions. This observation leads to the following definition:

Definition 4.1. For a sparse schedule $\mathcal{G}$, the **reduced-guidance path** is

$$\pi_t^{\mathcal{G}}(x_t) := \sum_{x_{\lceil t \rceil} \in \mathcal{X}} \pi_{\lceil t \rceil}^*(x_{\lceil t \rceil}) p_{t|\lceil t \rceil}(x_t|x_{\lceil t \rceil}). \tag{7}$$

A naive objective would be to find $\mathcal{G}$ such that only the final marginal $\pi_0^{\mathcal{G}}$ matches the target π_0^*. Theoretically, if the number of particles N is infinite, even a minimal schedule like $\mathcal{G} = \{0\}$ is sufficient to recover the exact reward-tilted distribution at the final step. However, this ignores a critical practical challenge: in the finite-N regime, delaying guidance until the end causes the sampler to postpone all reward information. This leads to diversity collapse and poor reward (see Fig. 1). To mitigate this degradation in practical sampling, we instead aim to minimize the deviation between the full-guidance and reduced-guidance paths across all timesteps.

Definition 4.2. The **Total Deviation** for a schedule $\mathcal{G}$ is defined as the average ℓ_1 distance between the full-guidance and reduced-guidance paths:

$$\mathbf{TD}(\mathcal{G}) := \frac{1}{T} \sum_{t=0}^{T-1} \|\pi_t^* - \pi_t^{\mathcal{G}}\|_1 = \frac{1}{T} \sum_{t=0}^{T-1} \sum_{x_t \in \mathcal{X}} |\pi_t^*(x_t) - \pi_t^{\mathcal{G}}(x_t)|. \tag{8}$$

This definition formalizes the core objective of VISTA: schedule selection is transformed from a heuristic choice of intervals into a principled, budgeted optimization to minimize the gap between the sparse sampling path and the ideal full-guidance probability path.

4.2 A TRACTABLE UPPER BOUND

A major obstacle in directly optimizing $\mathbf{TD}(\mathcal{G})$ is the intractability of the full-guidance path π_t^*. The following theorem provides a computable surrogate based on value expectations that can be estimated from a small number of warmup runs.

Theorem 4.3. *Assume $0 < R(x_0) \le R_{\max} < \infty$. For any schedule $\mathcal{G} \subset \mathcal{T}$,*

$$\mathbf{TD}(\mathcal{G}) \le \mathbf{UTD}(\mathcal{G}) := \frac{2}{T} \sum_{t \in \mathcal{T} \setminus \mathcal{G}} [1 - \frac{1}{R_{max}} \mathbb{E}_{x_{\lceil t \rceil} \sim \pi^*_{\lceil t \rceil}} [V_{\lceil t \rceil}(x_{\lceil t \rceil})]]. \tag{9}$$

The proof of this inequality is provided in Section A.3.

Our goal is to identify a schedule $\mathcal{G}$ that minimizes $\mathbf{UTD}$. This approach—optimizing a tractable upper bound when the primary objective is intractable—is a well-established strategy in machine learning Blei et al. (2017); Oord et al. (2018); Nowozin et al. (2016).

For a fixed budget of guidance steps $T' = |\mathcal{G}|$, minimizing $\mathbf{UTD}$ is equivalent to maximizing the following objective:

$$S(\mathcal{G}) := \sum_{t \in \mathcal{T} \setminus \mathcal{G}} \mathbb{E}_{x_{\lceil t \rceil} \sim \pi^*_{\lceil t \rceil}} \left[V_{\lceil t \rceil}(x_{\lceil t \rceil}) \right]. \tag{10}$$

Intuitively, if the state at the most recent active step $x_{\lceil t \rceil}$ already exhibits a high expected reward, subsequent unguided propagation is less likely to deviate significantly from the optimal path.

Crucially, this optimization does not require knowledge of the full marginals $\pi_t^*(x_t)$, nor the constant $R_{\max}$; it relies solely on the Monte Carlo estimation of $\mathbb{E}_{x_t \sim \pi_t^*} [V_t(x_t)]$.

4.3 DESCRIPTION OF THE METHOD

VISTA estimates $\mathbb{E}_{x_t \sim \pi_t^*} [V_t(x_t)]$ by running M full-guidance warmup simulations. Let $\hat{V}_t^{(i,m)}$ be the value estimate of i-th particle at timestep t during the m-th warmup run. We approximate the mean value as:

$$\mathbb{E}_{x_t \sim \pi_t^*} [V_t(x_t)] \approx \hat{V}_t := \frac{1}{MN} \sum_{m=1}^{M} \sum_{i=1}^{N} \hat{V}_t^{(i,m)}. \tag{11}$$

The empirical schedule $\hat{\mathcal{G}}$ is then obtained by solving:

$$\hat{\mathcal{G}} \in \arg \max_{\mathcal{G} \subset \mathcal{T} : |\mathcal{G}| = T'} \hat{S}(\mathcal{G}), \quad \hat{S}(\mathcal{G}) := \sum_{t \in \mathcal{T} \setminus \mathcal{G}} \hat{V}_{\lceil t \rceil} \tag{12}$$

The optimization in Eq. equation 12 is solved exactly by dynamic programming in $O(T^2 T')$ time (see Algorithm 6), which is negligible for the horizons used here. Once $\hat{\mathcal{G}}$ is selected, VISTA utilizes this sparse schedule for all subsequent generations to accelerate inference.

4.4 COMPUTATIONAL TIME ANALYSIS

VISTA requires M initial full-guidance generations for schedule selection. Once the schedule is determined, it has the same per-generation complexity as other sparse guidance methods under the same guidance budget. Table 2 summarizes the resulting computational costs.

5 EXPERIMENTS

To evaluate the effectiveness of VISTA, we focus our main experiments on Masked Diffusion Models (MDMs), with experiments on other diffusion models provided in Appendix B. We compare **VISTA**

Table 2: Per-generation cost after schedule selection. Here, p is a pretrained reverse-model evaluation, q is an amortized proposal, r is a reward-model evaluation, ∇v is a value-gradient evaluation, N is the number of particles, T is the diffusion horizon, T' is the guidance budget, and J is the number of rollout samples for value estimation (see Algorithm 5 for details).

Variant	Full Guidance	Sparse Guidance
SMC-BASE	$NT \times p,\ NTJ \times r$	$NT \times p,\ NT'J \times r$
SMC-GRAD	$NT \times p,\ NTJ \times r,\ NT \times \nabla v$	$NT \times p,\ NT'J \times r,\ NT' \times \nabla v$

against full guidance and several sparse scheduling baselines. Specifically, **Interval-k** ($k = 1, \ldots, 5$) allocates all guidance steps to the k-th equally sized segment of the reverse diffusion trajectory, where Interval-1 corresponds to the earliest denoising stage and Interval-5 to the latest. **Uniform** distributes the guidance steps evenly across the entire trajectory. **Top-V** selects the timesteps with the largest estimated values $\hat{V}_t$, while **Top-ΔV** selects the timesteps with the largest consecutive value differences $\hat{V}_t - \hat{V}_{t+1}$. Both value-based baselines use the same warmup value statistics as VISTA. Unless otherwise stated, all reported times correspond to the per-generation cost after schedule selection; warmup costs are reported separately. Comprehensive experimental details are available in Appendix B.

5.1 Synthetic Experiment: Binary MNIST

We first evaluate VISTA on Binary MNIST using the MDM with the SMC-grad proposal, based on the codebase of Ou et al. (2025). The reward is defined as $r(x_0) = \log p(c \mid x_0)$, where c denotes the condition "the digit is 4". We use a total horizon of $T = 100$ with $\alpha = 1.0$, $N = 20$ particles, $J = 1$ rollout for value estimation, and $M = 3$ full-guidance warmup generations. A sample x_0 is considered "valid" if $r(x_0) > -0.1$. To measure diversity, *Unique Valid* counts the number of distinct valid samples, where samples with Jaccard similarity ≥ 0.85 are treated as identical. Results are averaged over 50 independent seeds, yielding 1,000 generated samples per schedule. The warmup phase took 24.7 s. For evaluation, **UTD** was computed with $R_{\max} = 1$, which is valid since $R(x_0) = p(c \mid x_0) \leq 1$. Figure 2 summarizes the results, and complete numerical results, including the SMC-base setting, are provided in Appendix B.2.1.

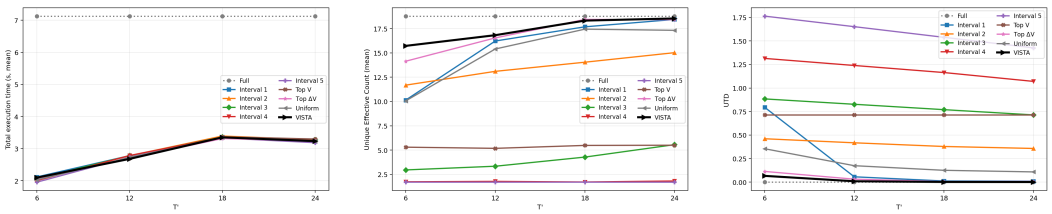


Figure 2: Binary MNIST (SMC-grad). Execution time, Unique Valid, and UTD as the guidance budget T' varies.

5.2 Toxic Text Generation

Toxic text generation is a high-dimensional discrete task where the reward model must guide the sampler against a strong language prior. The reward is defined as $r(x_0) = \log p(c \mid x_0)$ for a target toxicity condition. Without guidance, the base model satisfies the target toxicity condition in only 0.8% of generations. We report *Unique Toxic Ratio*, defined as the fraction of generated samples that are both toxic and unique, where samples with normalized Levenshtein distance ≤ 0.05 are treated as identical. We set $T = 100$, $\alpha = 0.2$, $N = 8$, $J = 4$, and $M = 3$, and use the SMC-grad proposal. For each of 15 prompts, we perform two independent runs, yielding $15 \times 2 \times 8 = 240$ generated samples for each scheduling method. The warmup phase required 639.6 s. Figure 3 summarizes the results, and complete numerical results are provided in Appendix B.2.2. Our implementation is based on the codebase of Ou et al. (2025).

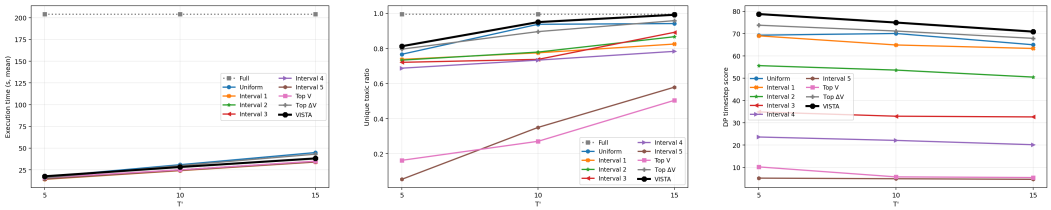


Figure 3: Toxic Text (SMC-grad). Execution time, Unique Toxic Ratio, and $\hat{S}(\mathcal{G})$ as the guidance budget T' varies.

5.3 CIFAR-10 Image Generation

We evaluate CIFAR-10 generation experiment using the MDM, SMC-base framework, based on the codebase of Schiff et al. (2024). The reward is defined as $r(x_0) = \log p(c|x_0)$, where c represents the condition "the image is an airplane". We use $T = 1,000$ time steps, $N = 20$ particles, $\alpha = 1.0$, $J = 5$ rollout for value estimation, and $M = 5$ warmup simulations. A sample x_0 is considered "valid" if $p(c|x_0) > 0.5$. *Unique Valid* is number of valid samples, where samples with LPIPS distance ≤ 0.03 are considered identical. The warmup phase took 217.9 s. Results are averaged over 50 runs, totaling $1,000$ generated samples per schedule. The graph of experiment result is in Figure 4.

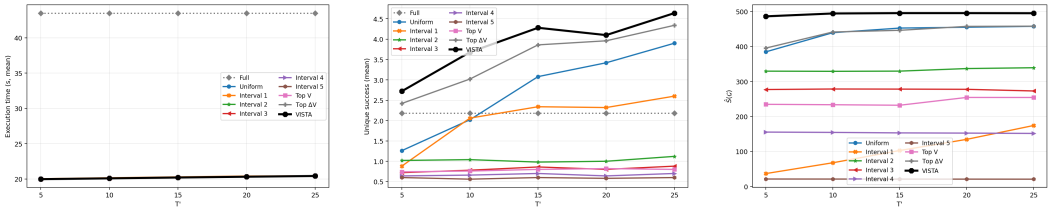


Figure 4: CIFAR-10 (SMC-base). Execution time, unique toxic samples, and $\hat{S}(\mathcal{G})$ as the guidance budget T' varies.

5.4 Biological Sequence Design

For DNA generation, we follow the experimental setup of (Ou et al., 2025). We use the reward $r(x_0) = \log p(c \mid x_0)$, where $p(c \mid x_0)$ denotes the predicted ATAC accessibility of the generated sequence. Following (Zekri & Boullé, 2026), a sequence is considered biologically active if its ATAC-Acc score exceeds 0.5. We define *Unique Valid* as the number of distinct biologically active sequences, where two sequences with Hamming distance ≤ 1 are treated as identical. We set $T = 128$, $\alpha = 0.01$, $N = 8$, $J = 4$, and $M = 3$. Results are averaged over 30 independent runs, yielding 240 generated sequences for each scheduling method. The warmup phase took 49.1 s.

For Protein Fitness, we use the pretrained masked diffusion model from (Sahoo et al., 2024) together with a fitness oracle as the reward model $r(x_0)$. A sequence is considered valid if $r(x_0) \geq 0.5$, and *Unique Valid* counts the number of distinct valid sequences, where two sequences with Hamming distance ≤ 1 are treated as identical. We set $T = 128$, $\alpha = 0.3$, $N = 32$, $J = 3$, and $M = 3$. Results are averaged over 20 independent runs, yielding 640 generated sequences for each scheduling method. the warmup phase took 19.5 s.

The main results are presented in Tables 3 and 4, with the corresponding schedules shown in Appendix B.2.4.

5.5 Discussion

The schedules selected by VISTA differ substantially across tasks, highlighting the importance of adapting guidance timing to the underlying model and reward function. Binary MNIST places most

Table 3: DNA sequence generation (SMC-grad): execution time, unique valid, and $\hat{S}(\mathcal{G})$.

T'	Policy	Time [s] ↓	Unique Valid ↑	$\hat{S}(\mathcal{G})$ ↑
128	Full	15.34 ± 0.97	$\mathbf{5.90 \pm 1.35}$	–
8	Interval 1	2.61 ± 0.63	0.23 ± 0.42	1.89×10^{-4}
	Interval 2	2.54 ± 0.54	0.20 ± 0.48	2.89×10^{-6}
	Interval 3	2.69 ± 0.66	0.60 ± 0.76	3.77
	Interval 4	2.87 ± 0.76	1.03 ± 1.62	4.57
	Interval 5	2.78 ± 0.75	1.27 ± 1.15	1.43×10^{-6}
	Top-V	2.66 ± 0.73	1.27 ± 1.15	1.43×10^{-6}
	Top-ΔV	2.94 ± 0.74	2.30 ± 1.99	8.95
	Uniform	2.79 ± 0.66	0.83 ± 0.52	7.77
	VISTA	2.84 ± 0.61	$\mathbf{2.73 \pm 1.90}$	**12.40**
24	Interval 1	5.04 ± 0.87	0.40 ± 0.61	2.64×10^{-5}
	Interval 2	5.09 ± 0.70	0.77 ± 1.02	0.15
	Interval 3	5.09 ± 0.72	1.50 ± 1.26	3.37
	Interval 4	4.96 ± 0.68	3.37 ± 1.85	2.66
	Interval 5	4.93 ± 0.84	3.20 ± 1.99	1.24×10^{-6}
	Top-V	5.05 ± 0.70	3.07 ± 1.41	0.19
	Top-ΔV	5.63 ± 0.80	$\mathbf{4.93 \pm 1.84}$	9.58
	Uniform	5.51 ± 0.80	2.70 ± 1.39	7.10
	VISTA	5.29 ± 0.81	4.17 ± 1.21	**12.08**

Table 4: Protein Fitness generation (SMC-base): execution time, unique valid, and $\hat{S}(\mathcal{G})$.

T'	Policy	Time [s] ↓	Unique Valid ↑	$\hat{S}(\mathcal{G})$ ↑
128	Full	3.12	$\mathbf{9.00 \pm 2.77}$	–
16	Interval 1	1.531 ± 0.003	13.20 ± 3.24	3.79×10^3
	Interval 2	1.530 ± 0.004	10.65 ± 4.44	2.95×10^3
	Interval 3	1.530 ± 0.002	8.10 ± 2.15	2.07×10^3
	Interval 4	1.529 ± 0.002	6.00 ± 2.20	1.07×10^3
	Interval 5	1.517 ± 0.003	5.35 ± 2.16	1.12×10^2
	Top-V	1.519 ± 0.002	8.40 ± 2.68	3.34×10^3
	Top-ΔV	1.531 ± 0.002	12.60 ± 2.58	4.23×10^3
	Uniform	1.521 ± 0.003	10.85 ± 3.79	3.83×10^3
	VISTA	1.530 ± 0.004	$\mathbf{14.15 \pm 3.80}$	$\mathbf{4.53 \times 10^3}$
32	Interval 1	1.777 ± 0.003	$\mathbf{12.60 \pm 4.04}$	3.57×10^3
	Interval 2	1.777 ± 0.005	9.60 ± 4.58	2.81×10^3
	Interval 3	1.776 ± 0.004	6.80 ± 3.24	1.82×10^3
	Interval 4	1.773 ± 0.005	6.00 ± 2.10	9.15×10^2
	Interval 5	1.760 ± 0.006	6.10 ± 2.47	96.0
	Top-V	1.762 ± 0.003	8.55 ± 2.95	2.62×10^3
	Top-ΔV	1.774 ± 0.008	11.25 ± 3.49	3.56×10^3
	Uniform	1.768 ± 0.004	11.80 ± 2.82	3.33×10^3
	VISTA	1.776 ± 0.002	12.00 ± 3.32	$\mathbf{3.94 \times 10^3}$

guidance steps in the early part of the reverse process, whereas Toxic Text, DNA, and CIFAR-10 distribute guidance more broadly across the trajectory. These distinct patterns suggest that a single fixed temporal schedule is unlikely to be uniformly effective across tasks.

Interestingly, sparse guidance can sometimes outperform full guidance on diversity-sensitive metrics, as observed on CIFAR-10. This is consistent with the possibility that overly frequent correction can reduce particle diversity in finite-particle SMC. Overall, the results indicate that selectively allocating guidance steps can provide a favorable trade-off between downstream performance and inference cost.

Finally, the weighted extension in Appendix B shows that the same schedule-selection principle can also be applied beyond masked diffusion through non-uniform temporal weighting.

6 Limitation

VISTA has three main limitations. First, VISTA requires a small number of full-guidance warmup generations before schedule optimization. These generations can be retained as model outputs while simultaneously providing the value statistics used for schedule selection, but they still introduce an upfront computational cost. Therefore, the benefit of VISTA is most pronounced when the selected schedule is reused across multiple subsequent generations.

Second, the UTD objective is a conservative upper bound on path deviation rather than a direct objective for downstream task performance. As a result, the schedule that optimizes $\hat{S}(\mathcal{G})$ does not necessarily maximize the downstream evaluation metric at every guidance budget. Tighter surrogates that better capture finite-particle effects and downstream sample quality are an important direction for future work.

Third, the weighted extension for non-masked diffusion requires a user-specified temporal weighting function $\lambda(t)$. Our experiments show that non-uniform weighting can extend VISTA beyond masked diffusion, but the appropriate choice of $\lambda(t)$ may depend on the diffusion process and task. Automatically learning or adapting these temporal weights remains an open problem.

7 Conclusion

We introduced VISTA, a training-free scheduler for sparse reward-guided SMC in discrete diffusion. By minimizing an estimable upper bound on path deviation, VISTA identifies task-specific guidance schedules under a fixed computational budget. Our experiments across image, text, and biological domains demonstrate that automatic schedule selection can retain strong downstream performance, and in several settings even outperform full guidance, while substantially reducing inference latency. Ultimately, this work shows that for efficient discrete diffusion alignment, choosing *when* to guide is as important as designing *how* to guide.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work for language editing, improving clarity and organization of the manuscript, and assisting with coding, debugging, and experimental workflow development. They were also used as an auxiliary tool for discussing possible analyses and presentation choices. All mathematical arguments, experimental designs, reported numerical results, literature claims, and final scientific conclusions were reviewed and verified by the authors. The authors take full responsibility for the content of the paper, including all text, code, analyses, and claims produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies an inference-time scheduling method and does not involve human-subject experiments or the collection of new personal data. Our experiments use previously released models, datasets, and reward functions. One evaluation considers toxic-text generation as a controlled stress test for reward-guided sampling. We recognize that making reward-guided generation more computationally efficient may also lower the cost of generating undesirable or harmful content. This experiment is included to evaluate the behavior of the scheduling method under a challenging reward-steering setting and is not intended to encourage harmful deployment. We do not release generated toxic-text samples, and we recommend that practical applications use appropriately designed reward functions together with application-specific safety filters, monitoring, and access controls. More generally, VISTA optimizes when a supplied reward is applied and therefore inherits limitations, biases, and potential misuse risks of the underlying generative and reward models.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce both the theoretical and empirical results throughout the paper and appendix. The main text specifies the VISTA objective, estimation procedure, dynamic-programming algorithm, computational-cost model, experimental metrics, and principal hyperparameters.

REFERENCES

Idan Achituve, Hai Victor Habi, Amir Rosenfeld, Arnon Netzer, Idit Diamant, and Ethan Fetaya. Inverse problem sampling in latent space using sequential monte carlo. *arXiv preprint arXiv:2502.05908*, 2025.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Angela Castillo, Jonas Kohler, Juan C Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1962–1970, 2025.

Lohithsai Yadala Chanchu, Hany Abdulsamad, and Christian A Naesseth. Discrete diffusion inference-time control with nested sequential monte carlo.

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

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. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Shengyu Feng, Xiang Kong, Shuang Ma, Aonan Zhang, Dong Yin, Chong Wang, Ruoming Pang, and Yiming Yang. Step-by-step reasoning for math problems via twisted sequential monte carlo. *arXiv preprint arXiv:2410.01920*, 2024.

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. *arXiv preprint arXiv:2502.11420*, 2025.

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.

Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

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

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *arXiv preprint arXiv:2506.20701*, 2025.

Byunghyun Kim, Minyoung Bae, and Jae-Gil Lee. Sample-efficient multi-round generative data augmentation for long-tail instance segmentation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Alignment without over-optimization: Training-free solution for diffusion models. *arXiv e-prints*, pp. arXiv–2501, 2025b.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025a.

Yunghee Lee, Byeonghyun Pak, Junwha Hong, and Hoseong Kim. Tortoise and hare guidance: Accelerating diffusion model inference with multirate integration. *arXiv preprint arXiv:2511.04117*, 2025b.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6804–6818, 2022.

Ziwei Luo, Ziqi Jin, Lei Wang, Lidong Bing, and Thomas B Schön. Self-rewarding sequential monte carlo for masked diffusion language models. *arXiv preprint arXiv:2602.01849*, 2026.

Zhengyao Lv, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. *arXiv preprint arXiv:2410.19355*, 2024.

Christian A Naesseth, Fredrik Lindsten, and Thomas B Schön. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):187–306, 2019.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zijing Ou, Chinmay Pani, and Yingzhen Li. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design. *arXiv e-prints*, pp. arXiv–2505, 2025.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

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. *arXiv preprint arXiv:2412.10193*, 2024.

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. *arXiv preprint arXiv:2501.06848*, 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024a.

Masatoshi Uehara, Yulai Zhao, Ehsan Hajiramezanali, Gabriele Scalia, Gokcen Eraslan, Avantika Lal, Sergey Levine, and Tommaso Biancalani. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. *Advances in Neural Information Processing Systems*, 37:127511–127535, 2024b.

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

Luhuan Wu, Yi Han, Christian A Naesseth, and John P Cunningham. Reverse diffusion sequential monte carlo samplers. *arXiv preprint arXiv:2508.05926*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Oussama Zekri and Nicolas Boullé. Fine-tuning discrete diffusion models with policy gradient methods. *Advances in Neural Information Processing Systems*, 38:152868–152906, 2026.

A Theoretical Details

A.1 Background

Lemma A.1.

$$V_t(x_t) = \mathbb{E}_{x_s \sim p_{s|t}(\cdot|x_t)}[V_s(x_s)] \tag{13}$$

where $0 \leq s < t \leq T$. *Also,*

$$\mathbb{E}_{x_t \sim p_t}[V_t(x_t)] = \mathbb{E}_{x_0 \sim p_0}[R(x_0)]. \tag{14}$$

Proof.

$$V_t(x_t) = \mathbb{E}_{x_0 \sim p_{0|t}(\cdot|x_t)}[R(x_0)] \tag{15}$$

comes from the definition of value function.

$$\mathbb{E}_{x_s \sim p_{s|t}(\cdot|x_t)}[V_s(x_s)] = \mathbb{E}_{x_s \sim p_{s|t}(\cdot|x_t)}[\mathbb{E}_{x_0 \sim p_{0|s}(\cdot|x_s)}[R(x_0)]] \tag{16}$$

$$= \mathbb{E}_{x_0 \sim p_{0|t}(\cdot|x_t)}[R(x_0)] = V_t(x_t). \tag{17}$$

$$\mathbb{E}_{x_T \sim p_T}[V_T(x_T)] = \mathbb{E}_{x_T \sim p_T}[\mathbb{E}_{x_0 \sim p_{0|T}(\cdot|x_T)}[R(x_0)]] = \mathbb{E}_{x_0 \sim p_0}[R(x_0)]. \tag{18}$$

□

Theorem A.2. $\{\pi_t\}_{t=T}^0$ *that maximizes*

$$\mathbb{E}_{\{\pi_t\}_{t=T}^0}[r(x_0) - \alpha \sum_{t=T-1}^{0} D_{\mathrm{KL}}(\pi_{t|t+1}(\cdot|x_{t+1})||p_{t|t+1}(\cdot|x_{t+1})) - \alpha D_{\mathrm{KL}}(\pi_T(\cdot)||p_T(\cdot))] \tag{19}$$

($\alpha > 0$*) is*

$$\pi_t(x_t) = \frac{1}{Z} V_t(x_t) p_t(x_t). \quad (Z := \mathbb{E}_{x_0 \sim p_0}[R(x_0)]) \tag{20}$$

Proof. We define the Markov process $\{\pi_t^*\}_{t=T}^0$ as

$$\pi_T^*(x_T) = \frac{1}{Z} V_T(x_T) p_T(x_T), \tag{21}$$

$$\pi_{t|t+1}^*(x_t|x_{t+1}) = \frac{V_t(x_t) p_{t|t+1}(x_t|x_{t+1})}{V_{t+1}(x_{t+1})} \quad (t = 0, 1, \cdots, T-1). \tag{22}$$

(These distributions are well-defined(normalized) by Lemma A.1.)

We observe

$$\mathbb{E}_{\{\pi_t\}_{t=T}^0}[r(x_0) - \alpha \sum_{t=T-1}^{0} D_{\mathrm{KL}}(\pi_{t|t+1}(\cdot|x_{t+1})||p_{t|t+1}(\cdot|x_{t+1})) - \alpha D_{\mathrm{KL}}(\pi_T(\cdot)||p_T(\cdot))] \tag{23}$$

$$= \mathbb{E}_{x_0 \sim \pi_0}[r(x_0)] - \alpha \sum_{t=T-1}^{0} \mathbb{E}_{x_t, x_{t+1} \sim \pi_{t,t+1}}[\log \frac{\pi_{t|t+1}(x_t|x_{t+1})}{p_{t|t+1}(x_t|x_{t+1})}] - \alpha \mathbb{E}_{x_T \sim \pi_T}[\log \frac{\pi_T(x_T)}{p_T(x_T)}] \tag{24}$$

$$= \mathbb{E}_{x_T \sim \pi_T} \mathbb{E}_{x_{T-1} \sim \pi_{T-1|T}(\cdot|x_T)} \cdots \mathbb{E}_{x_0 \sim \pi_{0|1}(\cdot|x_1)}$$

$$[r(x_0) - \alpha \sum_{t=T-1}^{0} \log \frac{\pi_{t|t+1}(x_t|x_{t+1})}{p_{t|t+1}(x_t|x_{t+1})} - \alpha \log \frac{\pi_T(x_T)}{p_T(x_T)}]. \tag{25}$$

If we use the recurrence relation

$$\mathbb{E}_{x_s \sim \pi_{s|s+1}(\cdot|x_{s+1})}[v_s(x_s) - \alpha \log \frac{\pi_{s|s+1}(x_s|x_{s+1})}{p_{s|s+1}(x_s|x_{s+1})}] \tag{26}$$

$$= \mathbb{E}_{x_s \sim \pi_{s|s+1}(\cdot|x_{s+1})}[v_{s+1}(x_{s+1}) - \alpha \log \frac{\pi_{s|s+1}(x_s|x_{s+1})}{\pi_{s|s+1}^*(x_s|x_{s+1})}] \tag{27}$$

$$= v_{s+1}(x_{s+1}) - \alpha D_{\mathrm{KL}}(\pi_{s|s+1}(\cdot|x_{s+1})||\pi_{s|s+1}^*(\cdot|x_{s+1})) \tag{28}$$

consequently, Eq. equation 25 could be simplified into

$$\mathbb{E}_{x_T\sim\pi_T}[v_T(x_T) - \alpha \log \frac{\pi_T(x_T)}{p_T(x_T)}] - \alpha \sum_{t=T-1}^{0} \mathbb{E}_{x_{t+1}\sim\pi_{t+1}}[D_{\mathrm{KL}}(\pi_{t|t+1}(\cdot|x_{t+1})||\pi^*_{t|t+1}(\cdot|x_{t+1}))] \tag{29}$$

$$=\alpha \log Z - \alpha \sum_{t=T-1}^{0} \mathbb{E}_{x_{t+1}\sim\pi_{t+1}}[D_{\mathrm{KL}}(\pi_{t|t+1}(\cdot|x_{t+1})||\pi^*_{t|t+1}(\cdot|x_{t+1}))] - \alpha D_{\mathrm{KL}}(\pi_T(\cdot)||\pi^*_T(\cdot)). \tag{30}$$

We now easily see that this objective is maximized iff $\pi_T = \pi^*_T$ and $\pi_{t|t+1} = \pi^*_{t|t+1}$ for all $t = 0, 1, \cdots, T-1$.

The final step is proving

$$\pi^*_t(x_t) = \frac{1}{Z} V_t(x_t) p_t(x_t). \tag{31}$$

This is satisfied at $t = T$. If

$$\pi^*_{t+1}(x_{t+1}) = \frac{1}{Z} V_{t+1}(x_{t+1}) p_t(x_{t+1}), \tag{32}$$

then

$$\pi^*_t(x_t) = \sum_{x_{t+1}} \pi^*_{t+1}(x_{t+1}) \pi^*_{t|t+1}(x_t|x_{t+1}) \tag{33}$$

$$= \sum_{x_{t+1}} \frac{V_{t+1}(x_{t+1}) p_{t+1}(x_{t+1})}{Z} \frac{V_t(x_t) p_{t|t+1}(x_t|x_{t+1})}{V_{t+1}(x_{t+1})} \tag{34}$$

$$= \frac{1}{Z} V_t(x_t) p_t(x_t). \tag{35}$$

Proof ends by induction.

□

Algorithm 3 Multinomial resampling

1: **Input:** $\{x^{(i)}\}_{i=1}^N, \{w^{(i)}\}_{i=1}^N$
2: **for** $k \leftarrow 1, 2, \cdots, N$ **do**
3: $\quad i_k \overset{\text{iid}}{\sim} \text{Cat}(i : \frac{w^{(i)}}{\sum_{n=1}^N w^{(n)}})$
4: $\quad y^{(k)} \leftarrow x^{(i_k)}$
5: **end for**
6: **Output:** $\{y^{(k)}\}_{k=1}^N, \{1\}_{k=1}^N$

A.2 Reduced SMC algorithm

Algorithm 3 defines the idealized multinomial resampling operator utilized in our analysis. In practice, more efficient variants such as systematic, adaptive, or partial resampling may be employed; we refer the reader to Naesseth et al. (2019) for a comprehensive review of these methods.

Lemma A.3 (Importance Sampling). *Let $(x^{(i)}, \tilde{x}^{(i)}) \overset{iid}{\sim} \rho(x, \tilde{x})$, and let $w(x, \tilde{x}) \geq 0$ be a weight function such that $w^{(i)} = w(x^{(i)}, \tilde{x}^{(i)})$. As $N \to \infty$, the output particles $\{y^{(k)}\}_{k=1}^N$ of Algorithm 3 satisfy:*

$$y^{(k)} \overset{iid}{\sim} \pi(x) \propto \sum_{\tilde{x}} w(x, \tilde{x}) \rho(x, \tilde{x}). \tag{36}$$

Algorithm 4 Full SMC guidance

1: **for** $i \leftarrow 1, 2, \cdots, N$ **do**
2: $\quad x_T^{(i)} \overset{\text{iid}}{\sim} q_T(x_T)$
3: $\quad x_0^{(i,j)} \overset{\text{iid}}{\sim} p_{0|T}(x_0|x_T^{(i)}) \quad (j = 1, 2, \cdots, J)$
4: $\quad \hat{V}_T^{(i)} \leftarrow \frac{1}{J}\sum_{j=1}^{J} R(x_0^{(i,j)})$
5: $\quad w_T^{(i)} \leftarrow p_T(x_T^{(i)}) \times \hat{V}_T^{(i)} / q_T(x_T^{(i)})$
6: **end for**
7: $\{x_T^{(i)}\}_{i=1}^N, \{w_T^{(i)}\}_{i=1}^N \leftarrow \text{Resample}(\{x_T^{(i)}\}_{i=1}^N, \{w_T^{(i)}\}_{i=1}^N)$
8: **for** $t \leftarrow T-1, T-2, \cdots, 0$ **do**
9: $\quad$ **for** $i \leftarrow 1, 2, \cdots, N$ **do**
10: $\quad\quad x_t^{(i)} \overset{\text{iid}}{\sim} q_{t|t+1}(x_t|x_{t+1}^{(i)})$
11: $\quad\quad x_0^{(i,j)} \overset{\text{iid}}{\sim} p_{0|t}(x_0|x_t^{(i)}) \quad (j = 1, 2, \cdots, J)$
12: $\quad\quad \hat{V}_t^{(i)} \leftarrow \frac{1}{J}\sum_{j=1}^{J} R(x_0^{(i,j)})$
13: $\quad\quad w_t^{(i)} \leftarrow \frac{p_{t|t+1}(x_t^{(i)}|x_{t+1}^{(i)})}{q_{t|t+1}(x_t^{(i)}|x_{t+1}^{(i)})} \times \frac{\hat{V}_t^{(i)}}{\hat{V}_{\lceil t+1 \rceil}^{(i)}} \times w_{\lceil t+1 \rceil}^{(i)}$
14: $\quad$ **end for**
15: $\quad \{x_t^{(i)}\}_{i=1}^N, \{w_t^{(i)}\}_{i=1}^N \leftarrow \text{Resample}(\{x_t^{(i)}\}_{i=1}^N, \{w_t^{(i)}\}_{i=1}^N)$
16: **end for**

Proof. For any test function $\phi(x)$,

$$\mathbb{E}_{x^{(i)},\tilde{x}^{(i)} \overset{\text{iid}}{\sim} \rho(x,\tilde{x})}[\phi(y^{(k)})] = \mathbb{E}_{x^{(i)},\tilde{x}^{(i)} \overset{\text{iid}}{\sim} \rho(x,\tilde{x})}[\frac{\frac{1}{N}\sum_{i=1}^{N} w(x^{(i)}, \tilde{x}^{(i)})\phi(x^{(i)})}{\frac{1}{N}\sum_{n=1}^{N} w(x^{(n)}, \tilde{x}^{(n)})}]. \tag{37}$$

By the law of large numbers, as $N \to \infty$,

$$\frac{1}{N}\sum_{n=1}^{N} w(x^{(n)}, \tilde{x}^{(n)}) \xrightarrow{\text{a.s.}} \mathbb{E}_{x,\tilde{x}\sim\rho}[w(x, \tilde{x})] \tag{38}$$

$$\frac{1}{N}\sum_{i=1}^{N} w(x^{(i)}, \tilde{x}^{(i)})\phi(x^{(i)}) \xrightarrow{\text{a.s.}} \mathbb{E}_{x,\tilde{x}\sim\rho}[w(x, \tilde{x})\phi(x)] = \mathbb{E}_{x,\tilde{x}\sim\rho}[w(x, \tilde{x})] \times \mathbb{E}_{x\sim\pi}[\phi(x)]. \tag{39}$$

Thus, we get

$$\mathbb{E}_{x^{(i)},\tilde{x}^{(i)} \overset{\text{iid}}{\sim} \rho(x,\tilde{x})}[\phi(y^{(k)})] \overset{N\to\infty}{\to} \mathbb{E}_{x\sim\pi}[\phi(x)], \tag{40}$$

so $y^{(k)} \sim \pi$.

Lastly, regarding the independence of $\{y^{(k)}\}_{k=1}^N$, the multinomial resampling process draws each $y^{(k)}$ independently given the particle set. As $N \to \infty$, this process is equivalent to i.i.d. sampling from the target distribution π, as the empirical distribution converges to the true π.

□

Theorem A.4. *Say $N, J \to \infty$. At every guided time $t \in \mathcal{G} \cup \{T\}$, the particles in Algorithm 5 are distributed according to the ideal guided target distribution, i.e., $x_t^{(i)} \overset{iid}{\sim} \pi_t^* \quad (i = 1, 2, \cdots, N)$.*

Proof. Since $J \to \infty$, the Monte Carlo value estimate $\hat{V}_t^{(i)}$ converges almost surely to the true value function $V_t(x_t^{(i)})$ by the strong law of large numbers.

Base Case: At $t = T$, the particles are sampled from q_T and weighted by $w_T \propto p_T(x_T)V_T(x_T)/q_T(x_T)$. By Lemma A.3, after resampling at Line 7, we have $x_T^{(i)} \overset{\text{iid}}{\sim} \pi_T^*(x_T) \propto p_T(x_T)V_T(x_T)$.

Inductive Step: Assume that $x_{\lceil t+1 \rceil}^{(i)} \overset{\text{iid}}{\sim} \pi_{\lceil t+1 \rceil}^*(x_{\lceil t+1 \rceil})$.

Algorithm 5 Sparse SMC guidance

1: **for** $i \leftarrow 1, 2, \cdots, N$ **do**
2: $\quad x_T^{(i)} \overset{\text{iid}}{\sim} q_T(x_T)$
3: $\quad x_0^{(i,j)} \overset{\text{iid}}{\sim} p_{0|T}(x_0|x_T^{(i)}) \quad (j = 1, 2, \cdots, J)$
4: $\quad \hat{V}_T^{(i)} \leftarrow \frac{1}{J}\sum_{j=1}^{J} R(x_0^{(i,j)})$
5: $\quad w_T^{(i)} \leftarrow p_T(x_T^{(i)}) \times \hat{V}_T^{(i)} / q_T(x_T^{(i)})$
6: **end for**
7: $\{x_T^{(i)}\}_{i=1}^N, \{w_T^{(i)}\}_{i=1}^N \leftarrow \text{Resample}(\{x_T^{(i)}\}_{i=1}^N, \{w_T^{(i)}\}_{i=1}^N)$
8: **for** $t \leftarrow T-1, T-2, \cdots, 0$ **do**
9: $\quad$ **if** $t \in \mathcal{G}$ **then**
10: $\quad\quad$ **for** $i \leftarrow 1, 2, \cdots, N$ **do**
11: $\quad\quad\quad x_t^{(i)} \overset{\text{iid}}{\sim} q_{t|t+1}(x_t|x_{t+1}^{(i)})$
12: $\quad\quad\quad x_0^{(i,j)} \overset{\text{iid}}{\sim} p_{0|t}(x_0|x_t^{(i)}) \quad (j = 1, 2, \cdots, J)$
13: $\quad\quad\quad \hat{V}_t^{(i)} \leftarrow \frac{1}{J}\sum_{j=1}^{J} R(x_0^{(i,j)})$
14: $\quad\quad\quad w_t^{(i)} \leftarrow \frac{p_{t|t+1}(x_t^{(i)}|x_{t+1}^{(i)})}{q_{t|t+1}(x_t^{(i)}|x_{t+1}^{(i)})} \times \frac{\hat{V}_t^{(i)}}{\hat{V}_{\lceil t+1 \rceil}^{(i)}} \times w_{\lceil t+1 \rceil}^{(i)}$
15: $\quad\quad$ **end for**
16: $\quad\quad \{x_t^{(i)}\}_{i=1}^N, \{w_t^{(i)}\}_{i=1}^N \leftarrow \text{Resample}(\{x_t^{(i)}\}_{i=1}^N, \{w_t^{(i)}\}_{i=1}^N)$
17: $\quad$ **else**
18: $\quad\quad$ **for** $i \leftarrow 1, 2, \cdots, N$ **do**
19: $\quad\quad\quad x_t^{(i)} \overset{\text{iid}}{\sim} p_{t|t+1}(x_t|x_{t+1}^{(i)})$
20: $\quad\quad$ **end for**
21: $\quad$ **end if**
22: **end for**

Then, just before the line 16,

$$(x_t^{(i)}, x_{t+1}^{(i)}, x_{\lceil t+1 \rceil}^{(i)}) \overset{\text{iid}}{\sim} q_{t|t+1}(x_t|x_{t+1}) \cdot p_{t+1|\lceil t+1 \rceil}(x_{t+1}|x_{\lceil t+1 \rceil}) \cdot \frac{p_{\lceil t+1 \rceil}(x_{\lceil t+1 \rceil}) V_{\lceil t+1 \rceil}(x_{\lceil t+1 \rceil})}{Z}. \tag{41}$$

Just after the line 16,

$$\begin{aligned}(x_t^{(i)}, x_{t+1}^{(i)}, x_{\lceil t+1 \rceil}^{(i)}) \overset{\text{iid}}{\sim}\ & q_{t|t+1}(x_t|x_{t+1}) p_{t+1|\lceil t+1 \rceil}(x_{t+1}|x_{\lceil t+1 \rceil}) \frac{p_{\lceil t+1 \rceil}(x_{\lceil t+1 \rceil}) V_{\lceil t+1 \rceil}(x_{\lceil t+1 \rceil})}{Z} \\ & \times \frac{p_{t|t+1}(x_t|x_{t+1})}{q_{t|t+1}(x_t|x_{t+1})} \frac{V_t(x_t)}{V_{\lceil t+1 \rceil}(x_{\lceil t+1 \rceil})} \qquad (42) \\ =\ & \frac{1}{Z} p_{t,t+1,\lceil t+1 \rceil}(x_t, x_{t+1}, x_{\lceil t+1 \rceil}) V_t(x_t). \qquad (43)\end{aligned}$$

By Lemma A.3, we may conclude that

$$x_t^{(i)} \overset{\text{iid}}{\sim} \pi_t^*(x_t). \tag{44}$$

The proof follows by induction for all $t \in \mathcal{G}$. □

A.3 Probability path distance

We first record a simple inequality.

Lemma A.5. *Random variable $X \in \mathbb{R}$ satisfies $0 \le X \le M$ and $\mu = \mathbb{E}[X]$. Then,*

$$\mathbb{E}[|X - \mu|] \le 2\mu \left(1 - \frac{\mu}{M}\right). \tag{45}$$

Proof. Since

$$0 = \mathbb{E}[X - \mu] = \mathbb{E}[(X - \mu)\mathbf{1}(X > \mu) + (X - \mu)\mathbf{1}(X \le \mu)],$$

we get the following identity:

$$\mathbb{E}[(X-\mu)\mathbf{1}(X>\mu)] = \mathbb{E}[(\mu-X)\mathbf{1}(X\le\mu)]. \tag{46}$$

Thus,

$$\begin{aligned}\mathbb{E}[|X-\mu|] &= \mathbb{E}[(X-\mu)\mathbf{1}(X>\mu)] \\ &\quad + \mathbb{E}[(\mu-X)\mathbf{1}(X\le\mu)] \\ &= 2\mathbb{E}[(\mu-X)\mathbf{1}(X\le\mu)] \\ &\le 2\mathbb{E}[\mu(1-X/M)] = 2\mu(1-\mu/M).\end{aligned}$$

Applying the expectation to both sides:

$$\begin{aligned}\mathbb{E}[|X-\mu|] &\le 2\mathbb{E}\left[\mu\left(1-\frac{X}{M}\right)\right] \\ &= 2\mu\left(1-\frac{\mathbb{E}[X]}{M}\right) = 2\mu\left(1-\frac{\mu}{M}\right).\end{aligned}$$

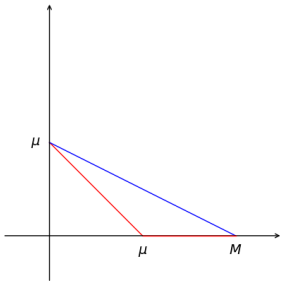


Figure 5: Illustration of $(\mu-X)\mathbf{1}(X\le\mu)\le\mu(1-\frac{X}{M})$

□

Proof of Theorem 4.3. If $t\in\mathcal{G}$, then $\pi_t^*=\pi_t^{\mathcal{G}}$, so

$$\mathbf{TD}(\mathcal{G}) = \frac{1}{T}\sum_{t\in\mathcal{T}-\mathcal{G}}\|\pi_t^*-\pi_t^{\mathcal{G}}\|_1. \tag{47}$$

By Lemma A.1,

$$V_{\lceil t\rceil}(x_{\lceil t\rceil}) = \mathbb{E}_{x_t\sim p_{t|\lceil t\rceil}(\cdot|x_{\lceil t\rceil})}[V_t(x_t)]. \tag{48}$$

Thus,

$$\|\pi_t^*-\pi_t^{\mathcal{G}}\|_1 = \sum_{x_t\in\mathcal{X}}|\pi_t^*(x_t) - \sum_{x_{\lceil t\rceil}\in\mathcal{X}}\pi_{\lceil t\rceil}^*(x_{\lceil t\rceil})p_{t|\lceil t\rceil}(x_t|x_{\lceil t\rceil})| \tag{49}$$

$$= \frac{1}{Z}\sum_{x_t\in\mathcal{X}}|\sum_{x_{\lceil t\rceil}\in\mathcal{X}}p_{\lceil t\rceil}(x_{\lceil t\rceil})p_{t|\lceil t\rceil}(x_t|x_{\lceil t\rceil})(V_t(x_t)-V_{\lceil t\rceil}(x_{\lceil t\rceil}))| \tag{50}$$

$$\le \frac{1}{Z}\mathbb{E}_{x_{\lceil t\rceil}\sim p_{\lceil t\rceil},x_t\sim p_{t|\lceil t\rceil}(\cdot|x_t)}|V_t(x_t)-V_{\lceil t\rceil}(x_{\lceil t\rceil})|. \tag{51}$$

Applying Lemma A.5 with $X=V_t(x_t)$ and $\mu=V_{\lceil t\rceil}(x_{\lceil t\rceil})$ gives

$$\mathbb{E}_{x_t\sim p_{t|\lceil t\rceil}(\cdot|x_{\lceil t\rceil})}[|V_t(x_t)-V_{\lceil t\rceil}(x_{\lceil t\rceil})|] \le 2V_{\lceil t\rceil}(x_{\lceil t\rceil})(1-\frac{V_{\lceil t\rceil}(x_{\lceil t\rceil})}{R_{\max}}). \tag{52}$$

Therefore,

$$\|\pi_t^*-\pi_t^{\mathcal{G}}\|_1 \le \frac{2}{Z}\mathbb{E}_{x_{\lceil t\rceil}\sim p_{\lceil t\rceil}}[V_{\lceil t\rceil}(x_{\lceil t\rceil})(1-\frac{V_{\lceil t\rceil}(x_{\lceil t\rceil})}{R_{\max}})] \tag{53}$$

$$= 2(1-\frac{1}{R_{\max}}\mathbb{E}_{x_{\lceil t\rceil}\sim\pi^*_{\lceil t\rceil}}[V_{\lceil t\rceil}(x_{\lceil t\rceil})]). \tag{54}$$

□

A.4 DP ALGORITHM

We aim to find $\mathcal{G}$ that maximizes

$$\hat{\mathbf{S}}(\mathcal{G}) := \sum_{t\in\mathcal{T}\setminus\mathcal{G}}\hat{V}_{\lceil t\rceil} \tag{55}$$

where $T' = |\mathcal{G}|$ is a given constant.

Define

$$dp[s][i] := \max_{\mathcal{G}'} \sum_{t \in \mathcal{T} \setminus \mathcal{G}'; t > s} \hat{V}_{\lceil t \rceil} \tag{56}$$

where

$$\mathcal{G}' = \{s = t_i < \cdots < t_1 < T\}. \tag{57}$$

We can observe the recurrence relation

$$dp[s][1] = (T - s - 1)\hat{V}_T \tag{58}$$

$$dp[s][i] = \max_{s < s' \leq T - i + 1} [dp[s'][i-1] + (s' - s - 1)\hat{V}_{s'}]. \quad (i \geq 2) \tag{59}$$

Thus, Algorithm 6 can compute the optimal $\mathcal{G}$.

Algorithm 6 DP algorithm

Initialize: $dp[s][i] \leftarrow 0$ for all s, i
for $s \leftarrow T' - 1, T', \cdots, T - 1$ **do**
    $dp[s][1] \leftarrow (T - s - 1)\hat{V}_T$
end for
for $i \leftarrow 2, 3, \cdots, T'$ **do**
    for $s \leftarrow T' - i, \cdots, T - i$ **do**
        for $s' \leftarrow s + 1$ to $T - i + 1$ **do**
            $c \leftarrow dp[s'][i-1] + (s' - s - 1)\hat{V}_{s'}$
            if $c > dp[s][i]$ **then**
                $dp[s][i] \leftarrow c$
                $parent[s][i] \leftarrow s'$
            end if
        end for
    end for
end for
$t_{T'} \leftarrow \text{argmax}_s(dp[s][T'] + s\hat{V}_s)$
for $j \leftarrow T' - 1, T' - 2, \cdots, 1$ **do**
    $t_j \leftarrow parent[t_{j+1}][j+1]$
end for
return $\mathcal{G} = \{t_1, t_2, \cdots, t_{T'}\}$

B Experimental Details

B.1 Experimental Setup

The Protein Fitness experiment was conducted on a single PRO 6000 GPU, while all other experiments were conducted on a single NVIDIA A6000 GPU. Unlike SMC-based evaluation protocols that generate N candidates but retain only the highest-reward sample, we retain all generated samples to evaluate both alignment and sample diversity.

For Binary MNIST, we use the masked diffusion model (MDM) from (Sahoo et al., 2024) and follow the implementation of (Ou et al., 2025). For Toxic Text Generation, we use the pretrained MDLM from (Sahoo et al., 2024) together with the reward model from (Logacheva et al., 2022), following the implementation of (Ou et al., 2025). For CIFAR-10, we use the MDM and implementation described in the corresponding experimental setup. For DNA generation, we follow the experimental setup of (Ou et al., 2025). For Protein Fitness, we use the pretrained MDLM from (Sahoo et al., 2024) together with a fitness oracle as the reward model. The resampling configurations for each experiment are summarized in Table 5.

Table 5: Resampling configurations for each experiment. Adaptive resampling is triggered when the normalized Effective Sample Size (ESS/N) falls below the specified threshold. Partial resampling denotes the number of particles replaced at each resampling step.

Experiment	ESS threshold	Partial resampling	Systematic resampling
Binary MNIST	0.95	$\lfloor N/2 \rfloor$	Yes
Toxic Text	0.98	N	Yes
CIFAR-10	0.95	$\lfloor N/2 \rfloor$	Yes
DNA Design	0.98	$\lfloor N/2 \rfloor$	Yes
Protein Fitness	0.98	$\lfloor N/2 \rfloor$	Yes
Continuous MNIST	0.98	$\lfloor N/2 \rfloor$	Yes

B.2 Details on main experiment

B.2.1 Details on Binary MNIST experiment

The warmup phase took 5.78s for SMC-base, and 24.7s for SMC-grad. See Figure 6, 7a, 7b and Table 6, 7.

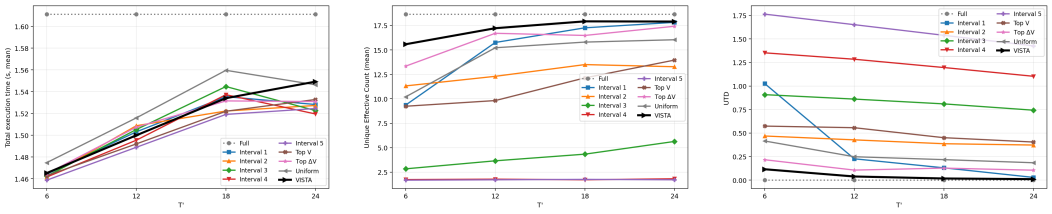
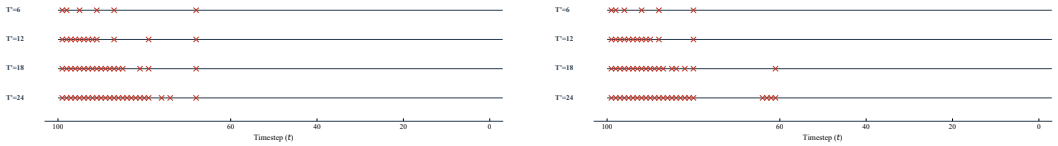


Figure 6: Binary MNIST (SMC-base). Execution time, unique valid samples, and UTD as the guidance budget T' varies.



(a) SMC-base (b) SMC-grad

Figure 7: VISTA schedules $\mathcal{G}$ for varying T' on Binary MNIST

B.2.2 Details on Toxic Text Generation

See Table 8 and Figure 8.

B.2.3 Details on CIFAR-10 Image Generation

See Table 9 and Figure 9.

“‘latex

B.2.4 Details on Biology Design

See Figure 10, 11.

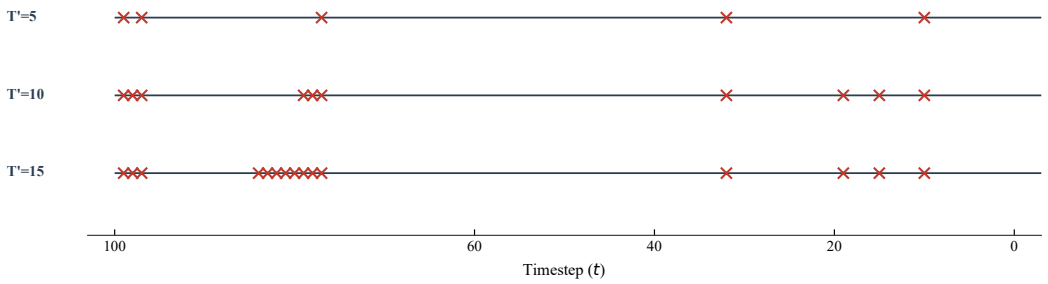


Figure 8: VISTA schedules $\mathcal{G}$ for varying T' (Toxic Text, SMC-grad)

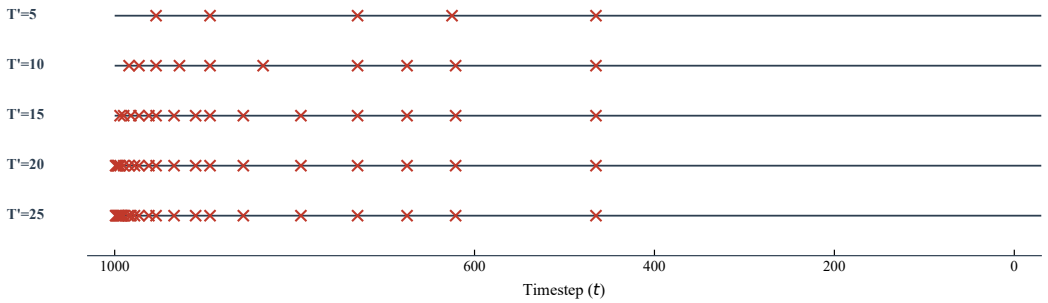


Figure 9: VISTA schedules $\mathcal{G}$ for varying T' (CIFAR-10, SMC-base)

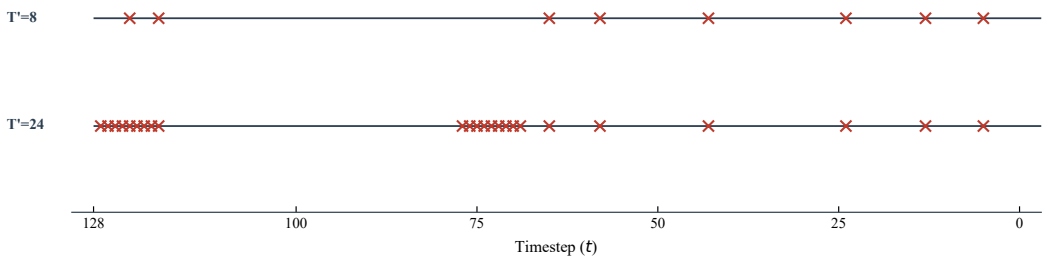


Figure 10: VISTA schedules $\mathcal{G}$ for varying T' (DNA, SMC-grad)

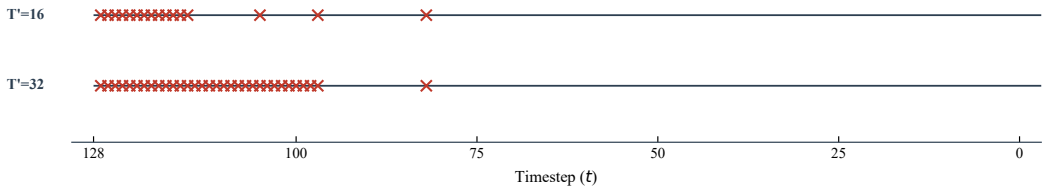


Figure 11: VISTA schedules $\mathcal{G}$ for varying T' (Protein Fitness, MDM, SMC-base)

Table 6: Binary MNIST (SMC-base): execution time, unique valid, and UTD.

T'	Policy	Time [s] ↓	Unique Valid ↑	UTD ↓
100	Full	1.61 ± 0.01	$\mathbf{18.66 \pm 1.78}$	–
6	Interval 1	1.46 ± 0.02	9.36 ± 3.53	1.026
	Interval 2	1.46 ± 0.02	11.32 ± 6.19	0.469
	Interval 3	1.46 ± 0.02	2.84 ± 2.61	0.908
	Interval 4	1.46 ± 0.02	1.74 ± 1.05	1.352
	Interval 5	1.46 ± 0.02	1.68 ± 1.04	1.763
	Top-V	1.46 ± 0.02	9.24 ± 5.84	0.574
	Top-ΔV	1.46 ± 0.02	13.34 ± 4.12	0.217
	Uniform	1.47 ± 0.09	10.20 ± 3.12	0.416
	VISTA	1.47 ± 0.02	$\mathbf{15.58 \pm 2.48}$	$\mathbf{0.116}$
12	Interval 1	1.50 ± 0.03	15.76 ± 3.54	0.227
	Interval 2	1.51 ± 0.11	12.30 ± 6.21	0.428
	Interval 3	1.51 ± 0.10	3.66 ± 3.47	0.861
	Interval 4	1.50 ± 0.04	1.78 ± 1.09	1.284
	Interval 5	1.49 ± 0.02	1.72 ± 1.05	1.651
	Top-V	1.49 ± 0.02	9.82 ± 5.94	0.557
	Top-ΔV	1.51 ± 0.07	16.70 ± 2.90	0.107
	Uniform	1.52 ± 0.12	15.22 ± 3.97	0.249
	VISTA	1.50 ± 0.02	$\mathbf{17.22 \pm 3.45}$	$\mathbf{3.91 \times 10^{-2}}$
18	Interval 1	1.54 ± 0.02	17.26 ± 3.33	0.131
	Interval 2	1.52 ± 0.02	13.50 ± 5.95	0.387
	Interval 3	1.54 ± 0.11	4.34 ± 4.03	0.810
	Interval 4	1.54 ± 0.08	1.72 ± 1.09	1.196
	Interval 5	1.52 ± 0.02	1.76 ± 1.08	1.538
	Top-V	1.52 ± 0.02	12.14 ± 6.14	0.451
	Top-ΔV	1.53 ± 0.03	16.48 ± 4.08	0.128
	Uniform	1.56 ± 0.16	15.80 ± 3.60	0.218
	VISTA	1.53 ± 0.04	$\mathbf{17.92 \pm 2.47}$	$\mathbf{1.79 \times 10^{-2}}$
24	Interval 1	1.53 ± 0.02	17.82 ± 2.60	2.93×10^{-2}
	Interval 2	1.53 ± 0.03	13.28 ± 5.82	0.373
	Interval 3	1.52 ± 0.02	5.64 ± 4.35	0.743
	Interval 4	1.52 ± 0.02	1.82 ± 1.16	1.103
	Interval 5	1.52 ± 0.03	1.72 ± 1.05	1.426
	Top-V	1.53 ± 0.10	13.96 ± 5.61	0.405
	Top-ΔV	1.53 ± 0.02	17.42 ± 3.00	0.106
	Uniform	1.55 ± 0.15	16.04 ± 3.69	0.185
	VISTA	1.55 ± 0.10	$\mathbf{17.90 \pm 3.05}$	$\mathbf{1.10 \times 10^{-2}}$

Table 7: Binary MNIST (SMC-grad): execution time, unique valid, and UTD.

T'	Policy	Time [s] ↓	Unique Valid ↑	UTD ↓
100	Full	7.13 ± 0.28	$\mathbf{18.78 \pm 1.73}$	–
6	Interval 1	2.12 ± 0.48	10.14 ± 3.93	0.795
	Interval 2	2.04 ± 0.36	11.68 ± 6.11	0.461
	Interval 3	2.07 ± 0.38	2.96 ± 2.57	0.884
	Interval 4	2.00 ± 0.33	1.74 ± 1.03	1.314
	Interval 5	1.96 ± 0.30	1.70 ± 1.07	1.763
	Top-V	2.09 ± 0.44	5.30 ± 4.12	0.714
	Top-ΔV	2.06 ± 0.44	14.14 ± 3.71	0.114
	Uniform	2.03 ± 0.42	10.06 ± 3.75	0.355
	VISTA	2.10 ± 0.47	$\mathbf{15.72 \pm 2.85}$	$\mathbf{6.69 \times 10^{-2}}$
12	Interval 1	2.78 ± 0.49	16.22 ± 2.97	5.70×10^{-2}
	Interval 2	2.78 ± 0.48	13.10 ± 6.36	0.418
	Interval 3	2.78 ± 0.48	3.34 ± 2.94	0.827
	Interval 4	2.79 ± 0.47	1.78 ± 1.07	1.239
	Interval 5	2.74 ± 0.48	1.70 ± 1.02	1.651
	Top-V	2.71 ± 0.52	5.18 ± 4.41	0.714
	Top-ΔV	2.70 ± 0.51	16.54 ± 2.84	3.19×10^{-2}
	Uniform	2.72 ± 0.53	15.40 ± 4.45	0.175
	VISTA	2.68 ± 0.52	$\mathbf{16.82 \pm 3.33}$	$\mathbf{9.72 \times 10^{-3}}$
18	Interval 1	3.37 ± 0.15	17.68 ± 2.79	1.26×10^{-2}
	Interval 2	3.39 ± 0.13	14.04 ± 5.99	0.379
	Interval 3	3.36 ± 0.21	4.28 ± 3.61	0.770
	Interval 4	3.35 ± 0.23	1.72 ± 1.03	1.164
	Interval 5	3.33 ± 0.27	1.70 ± 1.02	1.538
	Top-V	3.37 ± 0.26	5.48 ± 4.55	0.714
	Top-ΔV	3.31 ± 0.31	$\mathbf{18.44 \pm 1.67}$	6.34×10^{-3}
	Uniform	3.35 ± 0.42	17.44 ± 2.65	0.126
	VISTA	3.35 ± 0.24	18.32 ± 1.96	$\mathbf{2.25 \times 10^{-3}}$
24	Interval 1	3.23 ± 0.31	18.44 ± 2.04	1.03×10^{-2}
	Interval 2	3.28 ± 0.27	15.02 ± 5.06	0.358
	Interval 3	3.27 ± 0.28	5.56 ± 4.47	0.714
	Interval 4	3.25 ± 0.30	1.82 ± 1.35	1.070
	Interval 5	3.19 ± 0.30	1.70 ± 1.02	1.426
	Top-V	3.30 ± 0.30	5.50 ± 4.22	0.714
	Top-ΔV	3.24 ± 0.33	18.42 ± 1.69	7.43×10^{-3}
	Uniform	3.24 ± 0.40	17.32 ± 2.39	0.109
	VISTA	3.23 ± 0.30	$\mathbf{18.54 \pm 2.15}$	$\mathbf{1.43 \times 10^{-3}}$

B.3 Experiments beyond masked diffusion models

The weighted version of $\mathbf{TD}(\mathcal{G})$ is defined as

$$\mathbf{TD}(\mathcal{G}) := \sum_{t=0}^{T-1} \lambda(t) \|\pi_t^* - \pi_t^{\mathcal{G}}\|_1, \tag{60}$$

where $\lambda(t)$ is nonnegative weight. We used uniform weight in masked diffusion models. However, in uniform discrete diffusion or continuous diffusion models, if the state takes the wrong path at the beginning of sampling, it could be corrected later. Thus, we should give bigger weight when t is close to 0.

For the weighted version, by the proof of Theorem 4.3, the following inequality holds:

$$\sum_{t=0}^{T-1} \lambda(t) \|\pi_t^* - \pi_t^{\mathcal{G}}\|_1 \leq 2 \sum_{t \in \mathcal{T} \setminus \mathcal{G}} \lambda(t) [1 - \frac{1}{R_{max}} \mathbb{E}_{x_{\lceil t \rceil} \sim \pi^*_{\lceil t \rceil}} [V_{\lceil t \rceil}(x_{\lceil t \rceil})]] \tag{61}$$

$$\leq 2 \sum_{t=0}^{T-1} \lambda(t) - \frac{2}{R_{max}} \sum_{t \in \mathcal{T} \setminus \mathcal{G}} \lambda(t) \mathbb{E}_{x_{\lceil t \rceil} \sim \pi^*_{\lceil t \rceil}} [V_{\lceil t \rceil}(x_{\lceil t \rceil})]. \tag{62}$$

Thus, we aim to maximize the surrogate score

$$\hat{S}(\mathcal{G}) := \sum_{t \in \mathcal{T} \setminus \mathcal{G}} \lambda(t) \hat{V}_{\lceil t \rceil}, \tag{63}$$

in weighted VISTA, and the corresponding DP algorithm is modified into Alg 7. Further investigation on finding the optimal weight $\lambda(t)$ will be left to future work.

Table 8: Toxic text generation (SMC-grad).

T'	Policy	Time [s] ↓	Unique Toxic Ratio ↑	$\hat{S}(\mathcal{G})$ ↑
100	Full	204.23 ± 0.65	$\mathbf{1.00 \pm 0.02}$	**0**
5	Interval 1	14.10 ± 0.20	0.74 ± 0.34	68.97
	Interval 2	15.08 ± 0.32	0.73 ± 0.28	55.61
	Interval 3	15.01 ± 0.25	0.72 ± 0.35	34.85
	Interval 4	15.15 ± 0.35	0.69 ± 0.42	23.60
	Interval 5	14.34 ± 0.25	0.05 ± 0.17	5.16
	Top-V	16.01 ± 0.27	0.16 ± 0.30	10.20
	Top-ΔV	15.89 ± 0.27	0.80 ± 0.33	73.73
	Uniform	16.97 ± 0.27	0.77 ± 0.34	69.26
	VISTA	17.58 ± 0.25	$\mathbf{0.81 \pm 0.35}$	**78.73**
10	Interval 1	24.09 ± 0.24	0.78 ± 0.30	64.87
	Interval 2	25.07 ± 0.28	0.78 ± 0.31	53.61
	Interval 3	25.03 ± 0.24	0.74 ± 0.28	32.93
	Interval 4	25.01 ± 0.28	0.73 ± 0.37	22.08
	Interval 5	24.31 ± 0.27	0.35 ± 0.41	4.89
	Top-V	25.09 ± 0.28	0.27 ± 0.34	5.73
	Top-ΔV	29.93 ± 0.24	0.90 ± 0.21	71.12
	Uniform	30.94 ± 0.21	0.94 ± 0.09	70.02
	VISTA	28.28 ± 0.22	$\mathbf{0.95 \pm 0.11}$	**74.92**
15	Interval 1	33.98 ± 0.23	0.82 ± 0.20	63.30
	Interval 2	34.82 ± 0.20	0.87 ± 0.18	50.47
	Interval 3	34.76 ± 0.18	0.89 ± 0.18	32.61
	Interval 4	34.76 ± 0.24	0.78 ± 0.34	20.14
	Interval 5	34.09 ± 0.22	0.58 ± 0.42	4.62
	Top-V	35.02 ± 0.27	0.50 ± 0.42	5.43
	Top-ΔV	43.07 ± 0.21	0.96 ± 0.11	67.84
	Uniform	44.91 ± 0.19	0.94 ± 0.19	64.96
	VISTA	38.16 ± 0.24	$\mathbf{0.99 \pm 0.03}$	**70.86**

B.3.1 Protein Fitness (Uniform discrete diffusion)

We use $\lambda(t) = 1 - t/T$. We use reward model $r(x_0)$ as oracle that predicts fitness. We set $T = 128, N = 32, J = 3, M = 3, \alpha = 0.2$. A sample x_0 is considered "valid" if $r(x_0) \geq 0.5$. Two sequences with Hamming distance ≤ 1 are treated as identical. Results are averaged over 20 independent runs, yielding 640 generated samples per policy. The warmup phase took 18.7 s. See Table 10.

B.3.2 Continuous MNIST (Continuous diffusion)

We use $\lambda(t) = (1 - t/T)^{20}$. We use reward model $r(x_0) = \log p(c|x_0)$ where c indicates the condition 'image is 4'. We set $T = 100, N = 20, J = 1, M = 3, \alpha = 2.0$. A sample x_0 is considered "valid" if $r(x_0) > -0.1$. If the binarized images have Jaccard similarity ≥ 0.85, then we regard them as duplicates. Results are averaged over 30 independent runs, yielding 600 generated samples per policy. The warmup phase took 41.0 s. See Table 11.

Algorithm 7 DP algorithm

Initialize: $dp[s][i] \leftarrow 0$ for all s, i
for $s \leftarrow T' - 1, T', \cdots, T - 1$ **do**
    $dp[s][1] \leftarrow (\sum_{u>s} \lambda(u))\hat{V}_T$
end for
for $i \leftarrow 2, 3, \cdots, T'$ **do**
    for $s \leftarrow T' - i, \cdots, T - i$ **do**
        for $s' \leftarrow s + 1$ to $T - i + 1$ **do**
            $c \leftarrow dp[s'][i-1] + (\sum_{s<u<s'} \lambda(u))\hat{V}_{s'}$
            if $c > dp[s][i]$ **then**
                $dp[s][i] \leftarrow c$
                $parent[s][i] \leftarrow s'$
            end if
        end for
    end for
end for
$t_{T'} \leftarrow \text{argmax}_s(dp[s][T'] + (\sum_{u<s} \lambda(u))\hat{V}_s)$
for $j \leftarrow T' - 1, T' - 2, \cdots, 1$ **do**
    $t_j \leftarrow parent[t_{j+1}][j+1]$
end for
return $\mathcal{G} = \{t_1, t_2, \cdots, t_{T'}\}$

B.4 Sensitivity Analysis

Throughout the paper, we use M warmup simulations to estimate the value function for schedule optimization. Although this is a relatively small number of simulations, we find that it is sufficient to recover the high-level structure of the optimized guidance schedule. To verify this, we vary M and visualize the resulting schedules in Figures 12, 13, 14, 15, 16, 17, 18, and 19. Across all tasks, the value-critical guidance regions remain largely unchanged once $M = 3$ ($M = 5$ in CIFAR-10) is reached, indicating that only a small number of warmup simulations is required in practice.

C Societal Impacts

Positive impacts. VISTA reduces the cost of SMC-based reward guidance. This can make inference-time alignment more accessible to researchers with limited hardware and can reduce the energy cost of repeated reward-guided generation.

Potential negative impacts. More efficient control of generative models could be misused to generate harmful or manipulative content at lower cost. The toxic-text experiment is included only as a controlled benchmark for testing reward steering; it is not an endorsement of generating toxic content.

Mitigation. The main risk depends on the reward model and application. We recommend using VISTA with ethically aligned reward functions, safety filters, monitoring, and application-specific release controls. The scheduler itself is neutral: it decides when a given reward should be applied, but it does not define the reward.

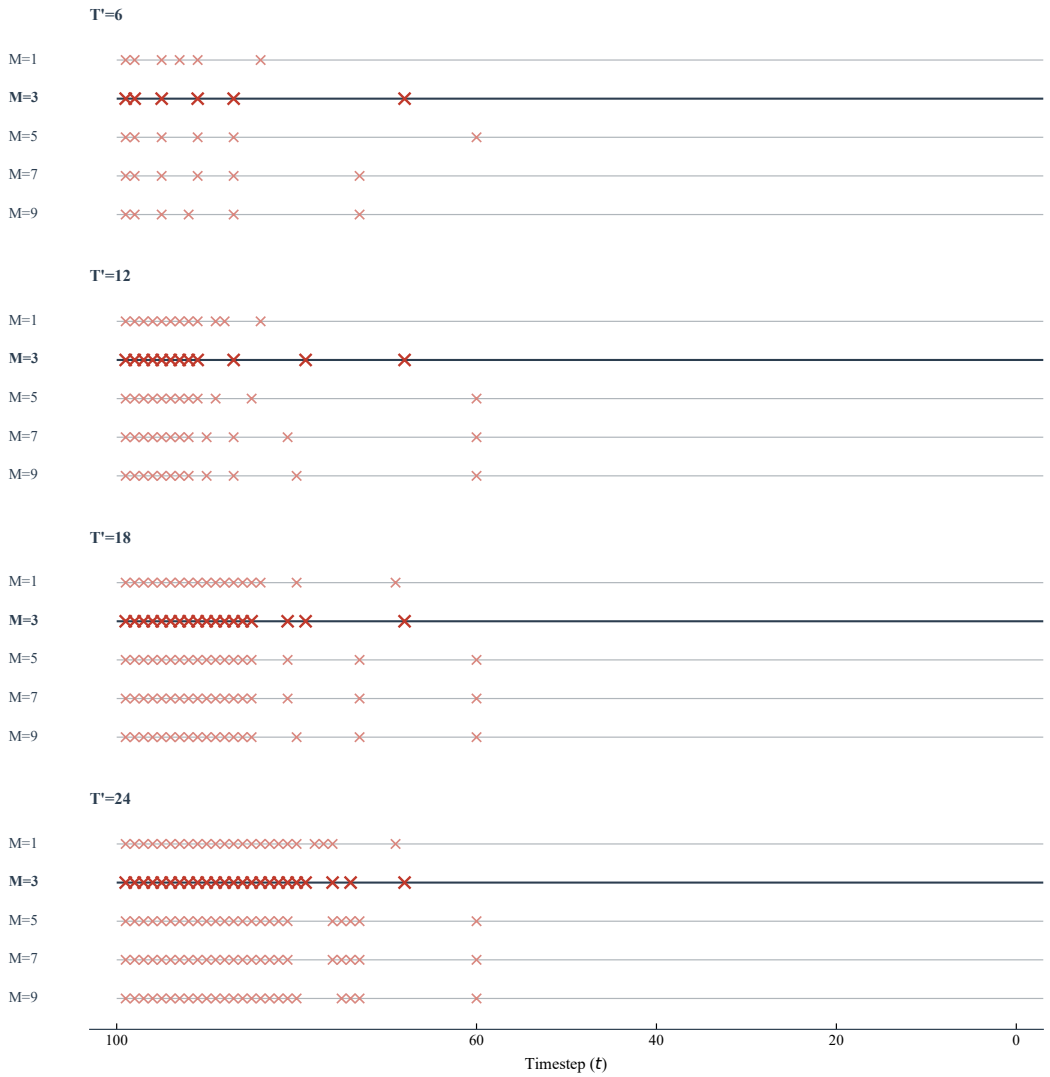


Figure 12: Optimized guidance schedules $\mathcal{G}$ for varying M (Binary MNIST, SMC-base)

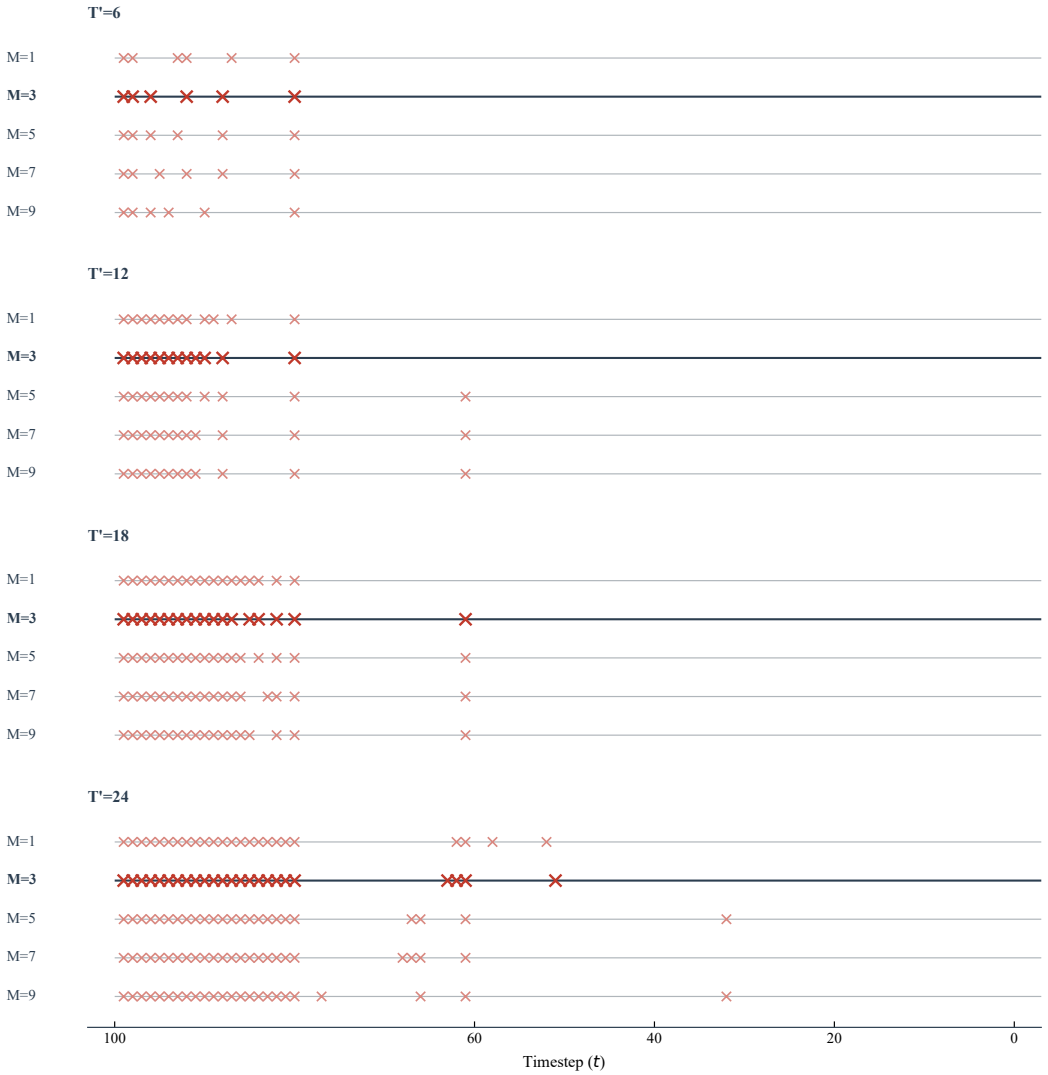


Figure 13: Optimized guidance schedules $\mathcal{G}$ for varying M (Binary MNIST, SMC-grad)

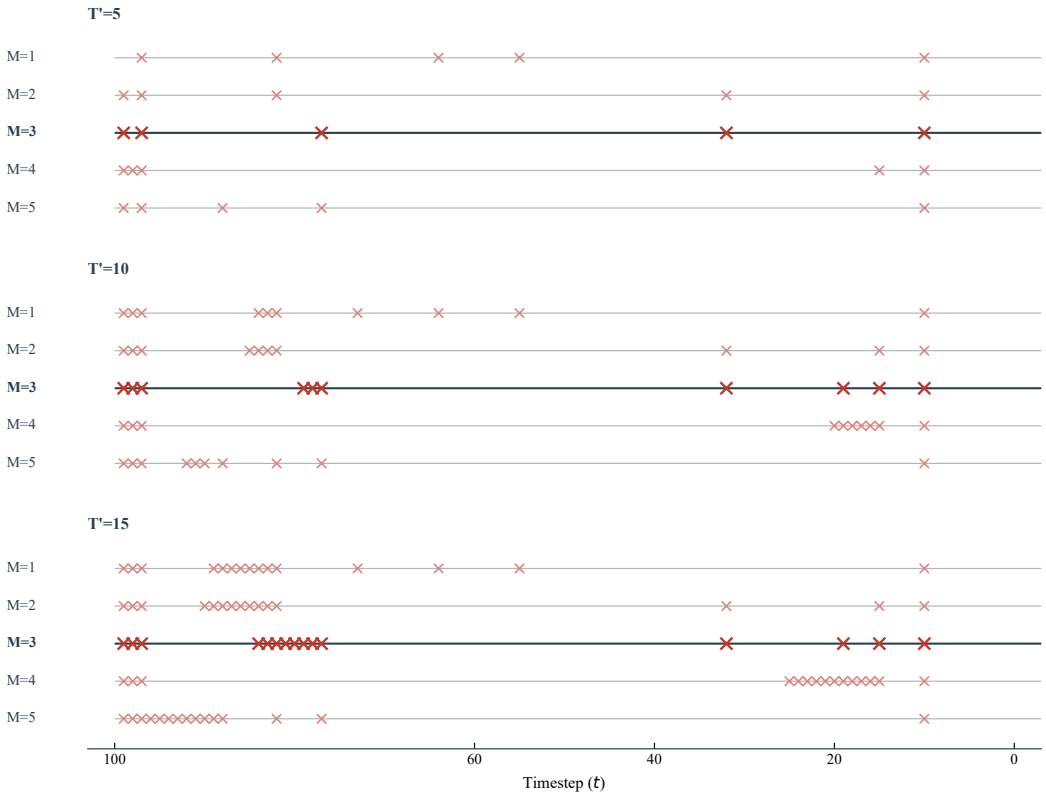


Figure 14: Optimized guidance schedules $\mathcal{G}$ for varying M (Toxic Text, SMC-grad)

Table 9: CIFAR-10 class-conditional generation: execution time, unique success count, and $\hat{S}(\mathcal{G})$.

T'	Policy	Time [s] ↓	Unique Valid ↑	$\hat{S}(\mathcal{G})$ ↑
1000	Full	43.51 ± 0.07	$\mathbf{2.18 \pm 1.02}$	$--$
5	Interval 1	20.03 ± 0.04	0.88 ± 0.46	36.77
	Interval 2	19.99 ± 0.03	1.02 ± 0.22	329.40
	Interval 3	19.98 ± 0.01	0.72 ± 0.26	276.90
	Interval 4	19.98 ± 0.02	0.64 ± 0.28	155.30
	Interval 5	19.98 ± 0.02	0.60 ± 0.30	21.21
	Top-V	19.98 ± 0.01	0.74 ± 0.32	234.60
	Top-ΔV	19.98 ± 0.01	2.42 ± 0.82	395.30
	Uniform	19.98 ± 0.01	1.26 ± 0.34	384.60
	VISTA	19.98 ± 0.01	$\mathbf{2.72 \pm 0.76}$	**486.00**
10	Interval 1	20.17 ± 0.03	2.06 ± 0.74	67.63
	Interval 2	20.09 ± 0.02	1.04 ± 0.28	328.90
	Interval 3	20.09 ± 0.02	0.78 ± 0.24	278.40
	Interval 4	20.09 ± 0.03	0.66 ± 0.30	154.40
	Interval 5	20.09 ± 0.02	0.56 ± 0.28	21.10
	Top-V	20.10 ± 0.01	0.76 ± 0.38	233.50
	Top-ΔV	20.10 ± 0.01	3.02 ± 1.38	441.30
	Uniform	20.09 ± 0.02	2.02 ± 0.46	439.30
	VISTA	20.09 ± 0.02	$\mathbf{3.68 \pm 1.02}$	**494.20**
15	Interval 1	20.29 ± 0.02	2.34 ± 0.94	102.70
	Interval 2	20.23 ± 0.02	0.98 ± 0.28	329.50
	Interval 3	20.22 ± 0.01	0.86 ± 0.36	278.10
	Interval 4	20.22 ± 0.03	0.70 ± 0.32	153.00
	Interval 5	20.22 ± 0.03	0.60 ± 0.26	20.99
	Top-V	20.22 ± 0.01	0.80 ± 0.36	232.00
	Top-ΔV	20.21 ± 0.01	3.86 ± 1.22	446.10
	Uniform	20.21 ± 0.03	3.08 ± 0.64	452.50
	VISTA	20.21 ± 0.02	$\mathbf{4.28 \pm 1.36}$	**495.20**
20	Interval 1	20.41 ± 0.03	2.32 ± 1.00	134.50
	Interval 2	20.34 ± 0.02	1.00 ± 0.48	336.80
	Interval 3	20.34 ± 0.03	0.80 ± 0.28	277.50
	Interval 4	20.34 ± 0.03	0.64 ± 0.30	152.40
	Interval 5	20.34 ± 0.02	0.58 ± 0.30	20.89
	Top-V	20.34 ± 0.02	0.82 ± 0.40	254.30
	Top-ΔV	20.34 ± 0.02	3.96 ± 1.30	457.30
	Uniform	20.34 ± 0.02	3.42 ± 0.72	454.90
	VISTA	20.32 ± 0.02	$\mathbf{4.10 \pm 1.12}$	**495.20**
25	Interval 1	20.46 ± 0.02	2.60 ± 1.14	174.10
	Interval 2	20.45 ± 0.02	1.12 ± 0.32	339.20
	Interval 3	20.45 ± 0.01	0.88 ± 0.26	272.80
	Interval 4	20.45 ± 0.02	0.70 ± 0.30	151.50
	Interval 5	20.44 ± 0.02	0.60 ± 0.32	20.78
	Top-V	20.46 ± 0.02	0.80 ± 0.38	254.20
	Top-ΔV	20.45 ± 0.02	4.34 ± 0.82	457.90
	Uniform	20.45 ± 0.02	3.90 ± 0.56	457.50
	VISTA	20.43 ± 0.02	$\mathbf{4.64 \pm 1.34}$	**494.90**

Table 10: Protein fitness generation, Uniform Discrete Diffusion (SMC-base, 32 particles $\times$ 20 runs): execution time, unique valid count, and $\hat{S}(\mathcal{G})$.

T'	**Policy**	**Time [s]** $\downarrow$	**Unique Valid** $\uparrow$	$\hat{S}(\mathcal{G})$ $\uparrow$
128	Full	3.26	$\mathbf{3.15 \pm 2.37}$	–
16	Interval 1	1.42	4.45 ± 3.19	2.71×10^3
	Interval 2	1.41	7.50 ± 3.72	6.05×10^3
	Interval 3	1.41	6.10 ± 3.93	7.86×10^3
	Interval 4	1.41	5.20 ± 2.84	3.53×10^3
	Interval 5	1.41	2.20 ± 1.20	602.24
	Top-V	1.42	5.95 ± 3.59	1.19×10^4
	Top-ΔV	1.41	6.40 ± 3.91	1.35×10^4
	Uniform	1.41	6.20 ± 3.90	1.10×10^4
	VISTA	1.42	$\mathbf{8.75 \pm 4.76}$	$\mathbf{2.06 \times 10^4}$
32	Interval 1	1.85	8.90 ± 5.31	1.07×10^4
	Interval 2	1.86	6.80 ± 4.64	6.02×10^3
	Interval 3	1.85	6.65 ± 4.32	6.08×10^3
	Interval 4	1.85	4.70 ± 3.34	4.35×10^3
	Interval 5	1.85	2.95 ± 2.50	443.11
	Top-V	1.85	5.60 ± 2.95	1.06×10^4
	Top-ΔV	1.85	6.05 ± 3.52	1.08×10^4
	Uniform	1.84	5.45 ± 2.14	8.88×10^3
	VISTA	1.85	$\mathbf{9.95 \pm 5.30}$	$\mathbf{1.97 \times 10^4}$

Table 11: MNIST (TDS, reproduce full/sparse): execution time, unique valid count, and $\hat{S}(\mathcal{G})$.

T'	**Policy**	**Time [s]** $\downarrow$	**Unique Valid** $\uparrow$	$\hat{S}(\mathcal{G})$ $\uparrow$
100	Full	15.47 ± 0.08	$\mathbf{10.33 \pm 3.45}$	–
6	Interval 1	4.29 ± 0.20	1.93 ± 1.44	0.06
	Interval 2	4.24 ± 0.03	1.60 ± 1.25	0.36
	Interval 3	4.24 ± 0.02	7.77 ± 4.64	3.40
	Interval 4	4.25 ± 0.03	12.87 ± 6.77	5.08
	Interval 5	4.15 ± 0.03	1.73 ± 1.28	0.11
	Top-V	4.26 ± 0.03	11.73 ± 5.91	4.92
	Top-ΔV	4.26 ± 0.03	14.13 ± 4.91	4.34
	Uniform	4.16 ± 0.02	14.10 ± 6.21	4.12
	VISTA	4.26 ± 0.03	$\mathbf{16.83 \pm 5.02}$	**5.11**
12	Interval 1	4.95 ± 0.37	1.93 ± 1.01	0.09
	Interval 2	4.86 ± 0.05	1.30 ± 1.09	1.21
	Interval 3	4.85 ± 0.04	13.43 ± 4.90	4.42
	Interval 4	4.86 ± 0.05	14.97 ± 5.57	5.06
	Interval 5	4.77 ± 0.05	1.87 ± 1.38	0.03
	Top-V	4.86 ± 0.37	8.20 ± 4.21	3.07
	Top-ΔV	4.90 ± 0.42	14.03 ± 5.40	4.67
	Uniform	4.75 ± 0.02	15.73 ± 5.72	3.96
	VISTA	5.31 ± 0.97	$\mathbf{17.13 \pm 3.80}$	**5.11**

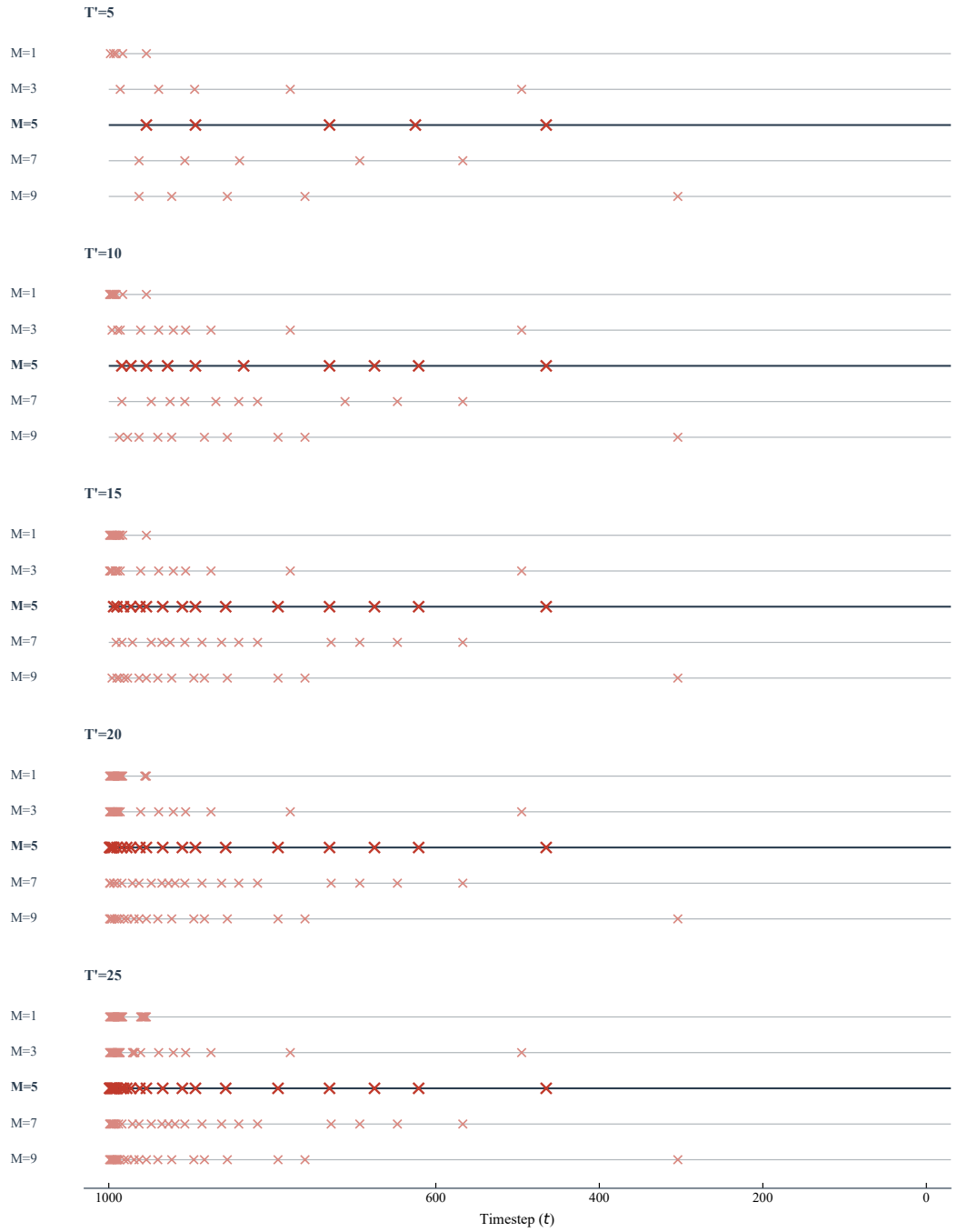


Figure 15: Optimized guidance schedules $\mathcal{G}$ for varying M (CIFAR-10, SMC-base)

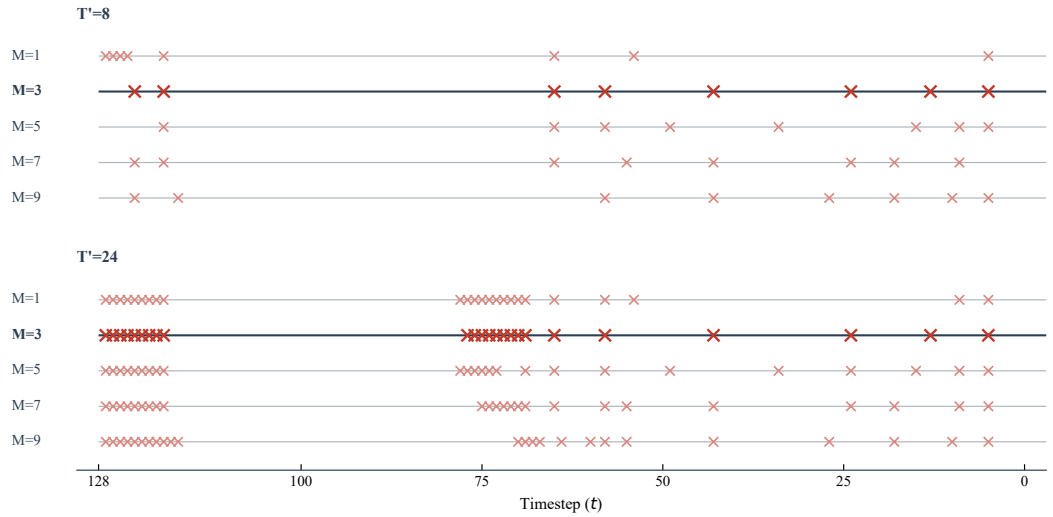


Figure 16: Optimized guidance schedules $\mathcal{G}$ for varying M (DNA, SMC-grad)

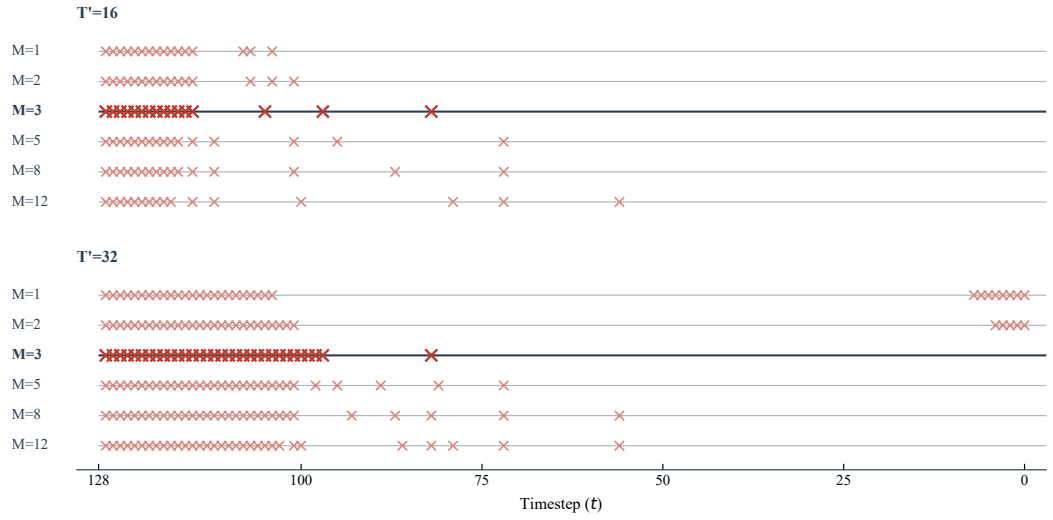


Figure 17: Optimized guidance schedules $\mathcal{G}$ for varying M (Protein Fitness, SMC-base, MDM)

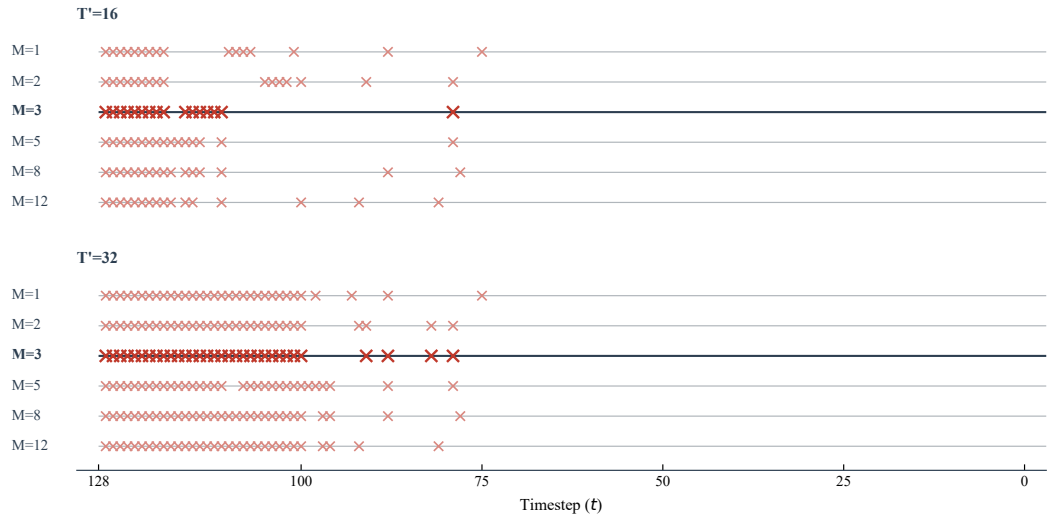


Figure 18: Optimized guidance schedules $\mathcal{G}$ for varying M (Protein Fitness, SMC-base, UDM)

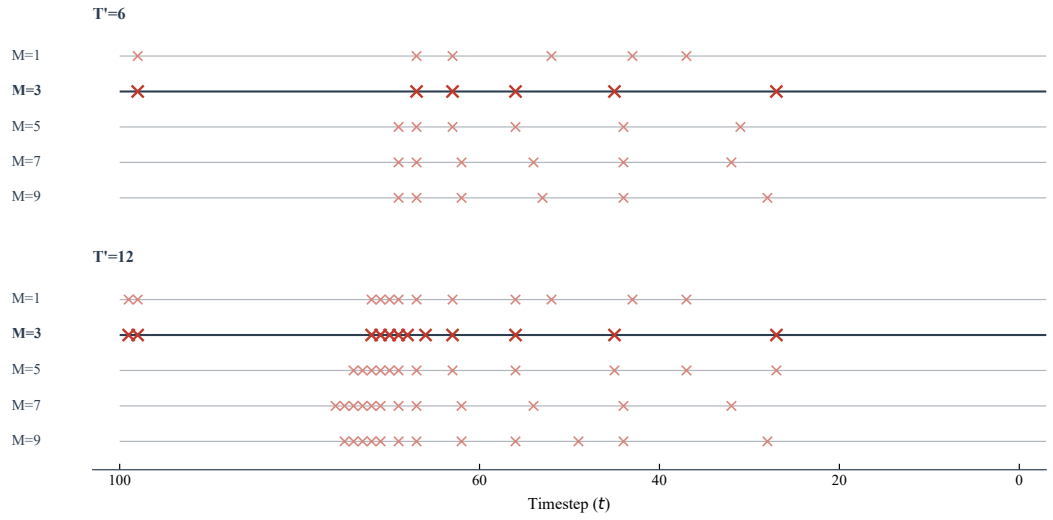


Figure 19: Optimized guidance schedules $\mathcal{G}$ for varying M (MNIST, TDS)