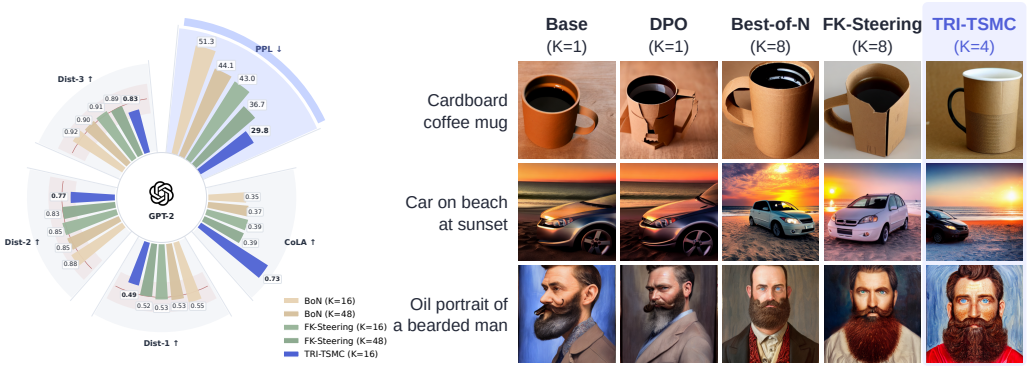
INFERENCE-TIME ALIGNMENT OF DIFFUSION MODELS VIA TRUST-REGION ITERATIVE TWISTED SEQUENTIAL MONTE CARLO

Anonymous authors
Paper under double-blind review

ABSTRACT

We study inference-time alignment of diffusion models: steering a fixed pretrained model toward high-reward outputs by changing its sampling procedure. Existing Sequential Monte Carlo (SMC) methods often rely on the base model to propose samples, then use reward information to reweight and resample them. When high-reward outputs are rare, this can lead to weight degeneracy and require many particles. Twisted SMC addresses this problem by modifying proposals using look-ahead information, but learning effective twisting functions can be unstable in high-dimensional diffusion models. We propose Trust-Region Iterative Twisted Sequential Monte Carlo (**TRI-TSMC**), which learns twisting functions through two steps. First, it constructs an intermediate trajectory distribution that moves toward the reward-tilted target while staying within a KL trust region around the current proposal. This step has a closed-form solution based on tempered importance weights. Second, it fits the twisting function to this distribution by weighted maximum likelihood. Theoretically, we characterize the optimal twist through a soft value function and show that it yields a zero-variance sampler. We prove that the exact trust-region update follows an escort path that reduces residual importance-weight variance, and that the population fitting objective is a forward-KL projection. Experiments on discrete diffusion text generation and text-to-image generation show improved primary alignment objectives under matched total sampled-trajectory budgets.



(a) Text alignment (MDLM) **(b) Image alignment (SD v1.5)**

Figure 1: **Text and image alignment with TRI-TSMC. (a)** Text generation on MDLM, evaluated using GPT-2 perplexity (PPL; ↓), grammatical acceptability (CoLA; ↑), and n-gram diversity (Dist-1/2/3; ↑). **(b)** Text-to-image generation on SD v1.5, with each row showing the same prompt across methods (columns). TRI-TSMC uses $3 \times 16 = 48$ sampled trajectories for text and $2 \times 4 = 8$ for images per output, matching the total sampled-trajectory budgets of Best-of-N and FK-Steering with $K = 48$ and $K = 8$, respectively.

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.

1 INTRODUCTION

We study inference-time alignment of diffusion models, where the goal is to generate high-reward outputs while keeping the pretrained model fixed. Existing alignment methods can be broadly grouped into three categories: (1) post-training fine-tuning methods; (2) distillation-based hybrid approaches; (3) inference-time alignment. Post-training methods, such as DDPPO and Diffusion-DPO (Black et al., 2024; Wallace et al., 2024), update the parameters of the pretrained base model to shift the learned sampling distribution toward high-reward outputs. They can produce efficient aligned samplers after training, but require access to model weights, reward-specific training data, and substantial training computation. They may also reduce sample diversity through alignment tax or mode collapse (Stiennon et al., 2020; Bai et al., 2022; Fan et al., 2023). Moreover, when the reward model or target preference changes, they usually require another round of optimization. Distillation methods reduce inference cost by training a student model to imitate a aligned sampler. They also require an additional training stage and heavily depend on the quality of the teacher sampler (Uehara et al., 2025). Inference-time alignment is more flexible because it keeps the pretrained base model fixed and only changes the sampling procedure. Gradient-based guidance methods, such as Classifier Guidance (Dhariwal & Nichol, 2021) and Diffusion Posterior Sampling (Chung et al., 2022), use local gradients from an external signal to adjust each denoising step. However, many alignment rewards are terminal, non-differentiable, or provided by black-box evaluators, such as preference models or image reward models. In these settings, there is no reliable local gradient at every intermediate denoising state. SMC-based steering methods address this issue by formulating alignment as sampling from a reward-tilted target distribution (Singhal et al., 2025; Wu et al., 2023a; Uehara et al., 2025). They can handle terminal or black-box rewards through particles, intermediate potentials, importance weighting, and resampling. However, their proposals often remain close to the base sampler. Reward information therefore acts mainly after particles have been generated. When high-reward trajectories are rare under the base model, a few particles can receive most of the weight, reducing particle efficiency. This leads to weight degeneracy, high-variance estimates, and a strong dependence on large particle budgets.

Twisted SMC uses look-ahead information to modify the proposal before sampling (Guarniero et al., 2017). A twisting function gives greater probability to states expected to lead to large future potentials. Theoretically, with the optimal twisting function, the residual importance weights become constant, giving a zero-variance importance sampler (Heng et al., 2020; Guarniero et al., 2017). The key challenge is to learn an effective twist from finite samples. Existing methods learn twists through recursive regression or path-space optimization (Guarniero et al., 2017; Lu & Wang, 2024), mainly developed for low-dimensional statistical inference and structured observation models. In contrast, diffusion alignment involves high-dimensional image, latent, or token states, neural reverse transitions, and rewards that are often terminal, black-box, or estimated from partially denoised samples. In this regime, recursive regression targets can be noisy, and direct path-space optimization can have high variance especially when the current proposal is far from the target.

In this work, we find that the optimal twist has a value-function interpretation: its logarithm satisfies a soft Bellman recursion under the Feynman-Kac formulation. This connects twist learning to value and policy learning in reinforcement learning, suggesting that twist learning could be stabilized by conservative local updates as in trust-region methods for RL and stochastic control problems (Schulman et al., 2015; Blessing et al., 2025). We propose **Trust-Region Iterative Twisted Sequential Monte Carlo (TRI-TSMC)**, which controls how much the proposal changes at each iteration. Specifically, TRI-TSMC first constructs an intermediate target within a KL trust region around the current proposal, then fits the twisting function to that target by weighted maximum likelihood.
Our main contributions are summarized as follows:

- We propose TRI-TSMC, a trust-region framework for iteratively learning twisting functions in SMC-based inference-time alignment of diffusion-based generative models. The method alternates between an exact KL-constrained path-space update and a weighted maximum-likelihood projection onto the parameterized twisted family.
- We provide a theoretical analysis of TRI-TSMC. We formalize the value-function interpretation of the optimal twisting function, show its zero-variance property, and prove that the trust-region update follows an escort path while reducing residual importance-weight variance.
- We evaluate TRI-TSMC on discrete diffusion text generation and text-to-image diffusion alignment. TRI-TSMC improves the alignment objectives over baselines under matched total sampled-trajectory budgets, with additional analyses showing the quality-diversity trade-off.

2 PRELIMINARIES

2.1 INFERENCE-TIME ALIGNMENT AS A UNIFIED SEQUENTIAL SAMPLING PROBLEM

We study inference-time alignment for diffusion-based generative models, including continuous-state diffusion models and discrete diffusion language models. We represent generation as a sequence of random states $X_{0:T} = (X_0, X_1, \dots, X_T)$, where $X_t \in \mathcal{X}_t$ and $\mathcal{X}_t$ denotes the respective state space. For continuous diffusion models, $\mathcal{X}_t$ is a continuous image or latent space; for discrete diffusion language models, $\mathcal{X}_t$ is a discrete token or mask-state space. The trajectory $X_{0:T}$ represents the full path generated by the reverse denoising process. For notational consistency, we index the process by the sampler iteration rather than the physical noise level; thus, X_0 denotes the initial state of the sampler and X_T denotes the final state at which the output is realized. The pretrained base model induces a base path measure $P(dx_{0:T}) = \mu(dx_0)\prod_{t=1}^{T} f_t(dx_t \mid x_{t-1})$, where μ is the initial distribution and f_t is the transition kernel from $\mathcal{X}_{t-1}$ to $\mathcal{X}_t$. Our method leverages the sequential nature of generation.

Inference-time alignment keeps the pretrained base model fixed and modifies only the sampling procedure. The base model induces the base path measure P, which serves as the reference distribution for alignment. Given a reward function $r : \mathcal{X}_T \to \mathbb{R}$ that measures a target attribute or human preference of the final output, the goal is to sample from a reward-tilted path distribution. Following Singhal et al. (2025); Uehara et al. (2025), we define the target distribution as

$$\pi(dx_{0:T}) \propto P(dx_{0:T}) \exp\left(\tfrac{r(x_T)}{\alpha}\right), \tag{2.1}$$

where $\alpha > 0$ is the temperature parameter, x_T denotes the final generated object, such as final clean sample in continuous diffusion or final token sequence in discrete diffusion language models.

A key challenge is that the reward in (2.1) is often terminal: it is evaluated only on the final output X_T, while intermediate states usually do not have a natural stepwise guidance signal. Thus direct sampling from π is difficult in long-horizon or high-dimensional generation, since most partial trajectories provide little information about whether they will eventually lead to a high-reward output.

2.2 SEQUENTIAL MONTE CARLO FOR TERMINAL-REWARD ALIGNMENT

To handle terminal rewards in a sequential way, we use the Feynman–Kac formulation. For notational simplicity, we use statewise potential functions $\{g_t(x_t)\}_{t=0}^{T}$ and define the target path distribution as

$$\pi(dx_{0:T}) = Z^{-1}\gamma(dx_{0:T}) = Z^{-1}P(dx_{0:T})\prod_{t=0}^{T} g_t(x_t). \tag{2.2}$$

where $\gamma(dx_{0:T})$ is the unnormalized target path measure, and $Z = \mathbb{E}_{X\sim P}[\prod_{t=0}^{T} g_t(X_t)]$ is the normalizing constant. The terminal-reward alignment target in (2.1) is a special case of (2.2), if

$$g_t(x_t) = 1, \quad t = 0, \dots, T-1, \qquad g_T(x_T) = \exp(r(x_T)/\alpha),$$

For a proposal measure Q such that $\gamma \ll Q$, we use the unnormalized importance weight

$$w(X_{0:T}) = \tfrac{d\gamma}{dQ}(X_{0:T}), \qquad \tfrac{d\pi}{dQ}(X_{0:T}) = \tfrac{1}{Z}w(X_{0:T}). \tag{2.3}$$

The importance weight satisfies $\mathbb{E}_{X\sim Q}[w(X_{0:T})] = Z$. Standard SMC approximates π with a system of K particles that are propagated, weighted, and resampled over time (refer to Appendix B for further details). In the standard setting, the proposal is the base model itself, i.e., $Q = P$, and

$$w(X_{0:T}) = \tfrac{d\gamma}{dP}(X_{0:T}) = \prod_{t=0}^{T} g_t(X_t). \tag{2.4}$$

Recent SMC-based steering methods for diffusion models define intermediate potentials based on reward estimates and apply particle resampling during generation (Singhal et al., 2025; Wu et al., 2023a). These methods are appealing because they are training-free and capable of handling terminal or black-box rewards.

However, traditional SMC-based steering propagates particles predominantly according to the base model's dynamics. When the reward-tilted target distribution is substantially distant from the base model's distribution, many particles are initially sampled from low-reward regions and only corrected after propagation via importance weighting and resampling. In long-horizon or high-dimensional generation tasks, this approach leads to poor particle efficiency, weight degeneracy, and high-variance estimates. This fundamental inefficiency motivates modifying the proposal itself using look-ahead information, the core principle behind Twisted SMC.

3 METHODOLOGY: TRI-TSMC

3.1 TWISTED SEQUENTIAL MONTE CARLO FOR DIFFUSION ALIGNMENT

The key limitation of standard SMC is that particle propagation proceeds following the base dynamics before the resulting importance weights are calculated and applied for correction. Twisted Sequential Monte Carlo (TSMC) addresses this by modifying the proposal distribution itself. Given the Feynman-Kac target measure $\pi(dx_{0:T}) = Z^{-1}\gamma(dx_{0:T}) \propto P(dx_{0:T}) \prod_{t=0}^{T} g_t(x_t)$, TSMC introduces a sequence of positive twisting functions $\psi = \{\psi_t(\cdot)\}_{t=0}^{T}$. Intuitively, $\psi_t(x_t)$ provides look-ahead information on the likelihood of the current state x_t leading to high future potentials.

Given the base process $P(dx_{0:T}) = \mu(dx_0) \prod_{t=1}^{T} f_t(dx_t \mid x_{t-1})$, the twisted initial distribution and twisted transition kernels are

$$\mu^{\psi}(dx_0) = \frac{\psi_0(x_0)}{c_\psi}\mu(dx_0), \qquad f_t^{\psi}(dx_t \mid x_{t-1}) = \frac{\psi_t(x_t)}{\tilde{\psi}_{t-1}(x_{t-1})} f_t(dx_t \mid x_{t-1}), \tag{3.1}$$

where $c_\psi = \int \psi_0(x_0)\mu(dx_0)$, $\tilde{\psi}_{t-1}(x_{t-1}) = \int \psi_t(x_t) f_t(dx_t \mid x_{t-1})$, $t = 1, \ldots, T$, and we set $\tilde{\psi}_T(\cdot) \equiv 1$. Let P^{ψ} denote the induced twisted path measure. The objective of twist learning is to choose ψ so that P^{ψ} becomes a better proposal for the Feynman-Kac target: particles should be guided toward trajectories with large future potentials before being corrected by importance weights. The residual importance weight corrects the mismatch between the twisted proposal and the target:

$$w^{\psi}(X_{0:T}) = \frac{d\gamma}{dP^{\psi}}(X_{0:T}) = c_\psi \prod_{t=0}^{T} g_t(X_t) \frac{\tilde{\psi}_t(X_t)}{\psi_t(X_t)}. \tag{3.2}$$

Thus, a well-chosen twisting function reduces the variance of this residual importance weight, thereby improving particle efficiency and bringing the particle approximation closer to the target distribution. We provide a detailed decomposition of this incremental weight in Appendix B.

Theoretically, Guarniero et al. (2017) showed that an optimal twisting function, ψ^*, exists that renders the residual importance weight constant, achieving a zero-variance importance sampler. This optimal function satisfies the backward recursion:

$$\psi_t^*(x_t) = g_t(x_t) \int \psi_{t+1}^*(x_{t+1}) f_{t+1}(dx_{t+1} \mid x_t), \qquad t = 0, \ldots, T-1, \tag{3.3}$$

with terminal condition $\psi_T^*(x_T) = g_T(x_T)$. (3.3) immediately implies $\psi_t^*(x_t) = g_t(x_t)\tilde{\psi}_t^*(x_t)$ by definition. In particular, in the terminal-reward case, we have $\psi_t^*(x_t) = \mathbb{E}_P[\exp(r(X_T)/\alpha) | X_t = x_t]$. The optimal twist provides a look-ahead value of the current denoising state, and the twisted transition favors next states with larger expected future potential. If $\psi = \psi^*$, the residual importance weight becomes constant and the twisted proposal exactly matches the target path distribution. Since computing ψ^* exactly is intractable in high-dimensional generation, the central algorithmic question is how to learn a parameterized twisting function $\psi(\theta)$ stably from finite samples.

3.2 ALGORITHM DESIGN

Existing SMC-based approaches, such as iAPF (which approximates the optimal twisting functions via recursive regression on simulated particles (Guarniero et al., 2017)) and TPPF (which directly optimizes neural twisting functions through a path-space KL objective (Lu & Wang, 2024)), are effective in simpler sequential statistical inference settings. However, applying them to the high-dimensional, nonlinear dynamics of generative model alignment can introduce instability. Specifically, direct path-space optimization often suffers from high variance when the current proposal is far from the target distribution, complicating twist learning during early training stages.

We propose **Trust-Region Iterative Twisted Sequential Monte Carlo (TRI-TSMC)**, summarized in Algorithm 1 and illustratted in Figure 2. The key idea is to decouple twist learning into two stages. First, it constructs an intermediate distribution over trajectories. This distribution moves toward the target but stays within a KL trust region around the current proposal. Second, it fits the parameterized twisting function to the intermediate distribution by weighted maximum likelihood.

Trust Region Update in Path Space. Let $\psi(\theta_i) = \{\psi_t(\cdot; \theta_i)\}_{t=0}^{T}$ denote the parameterized twisting functions at outer iteration i, and let $P_i = P^{\psi(\theta_i)}$ be the corresponding twisted path measure induced by (3.1). We define the normalized density ratio as follows,

$$R_i(X_{0:T}) := \frac{d\pi}{dP_i}(X_{0:T}) = \frac{1}{Z} w^{\psi(\theta_i)}(X_{0:T}), \tag{3.4}$$

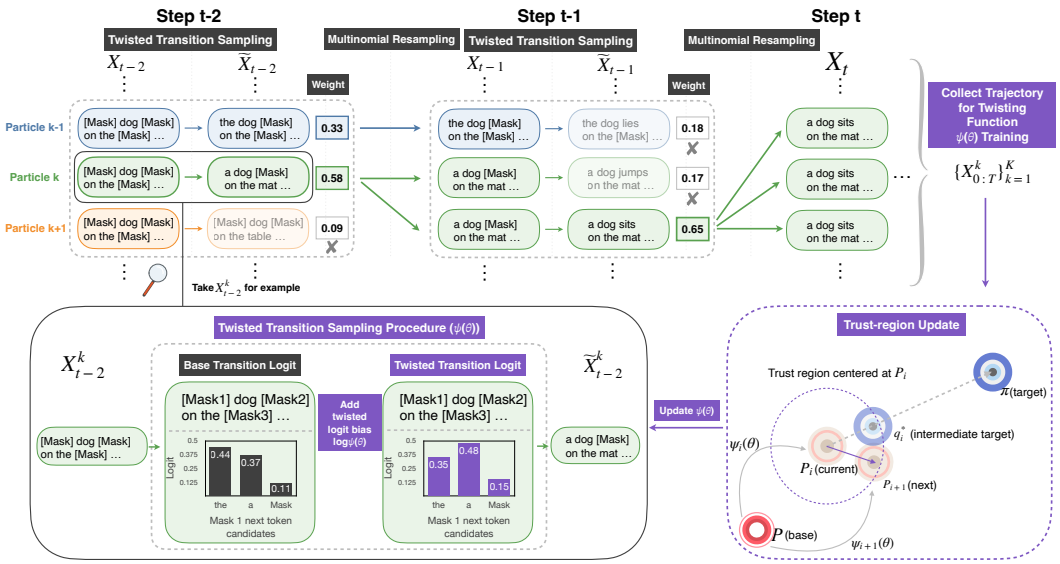


Figure 2: Overview of TRI-TSMC. The sampler alternates between twisted transition sampling and multinomial resampling to propagate particles toward high-reward trajectories. The twisting function modifies the base transition by adding a learned logit bias, providing look-ahead guidance during generation. Collected trajectories are then used in a trust-region update, which computes tempered trajectory weights and trains the next twisting function by weighted maximum likelihood.

where $w^{\psi(\theta_i)}$ is the unnormalized importance weight in (3.2). Our goal is to find an intermediate distribution q that is close to the target distribution π, while remaining within a trust region around the current path measure P_i. We formulate this as

$$q_i^* = \operatorname{argmin}_q D_{\mathrm{KL}}(q\|\pi) \quad \text{s.t.} \quad D_{\mathrm{KL}}\big(q\|P_i\big) \le \epsilon. \tag{3.5}$$

The radius $\epsilon > 0$ controls how far the intermediate distribution can move from the current proposal.

The next lemma shows that this trust-region update has a closed-form solution and achieves a monotone improvement in distribution space.

Lemma 3.1 (Trust-region path-space update)**.** Let $P_i = P^{\psi(\theta_i)}$ be the current twisted path measure, and let q_i^* be the solution to (3.5). Then there exists $\lambda_i \ge 0$ such that

$$\frac{dq_i^*}{dP_i}(X_{0:T}) \propto \big(R_i(X_{0:T})\big)^{\frac{1}{1+\lambda_i}} \propto \big(w^{\psi(\theta_i)}(X_{0:T})\big)^{\frac{1}{1+\lambda_i}}. \tag{3.6}$$

Moreover, we have $D_{\mathrm{KL}}(q_i^*\|\pi) \le D_{\mathrm{KL}}(P_i\|\pi)$.

Efficient Implementation via Weighted Maximum Likelihood. In practice, we evaluate the empirical dual using the ancestral trajectories $\{\xi^k\}_{k=1}^K$ returned by the current twisted-SMC sweep.

$$\hat\lambda_i = \operatorname{argmin}_{\lambda\ge 0}\left[\lambda\epsilon + (1+\lambda)\log\left(\tfrac{1}{K}\textstyle\sum_{k=1}^K \big(w^{\psi(\theta_i)}(\xi^k)\big)^{\frac{1}{1+\lambda}}\right)\right]. \tag{3.7}$$

By (3.6), the tempered regression weights are $\hat w_k = (w^{\psi(\theta_i)}(\xi^k))^{\hat\tau_i}/\sum_{j=1}^K (w^{\psi(\theta_i)}(\xi^j))^{\hat\tau_i}$, where $\hat\tau_i = 1/(1+\hat\lambda_i)$. We then fit the twisting function by weighted maximum likelihood:

$$\mathcal{L}(\theta) = -\textstyle\sum_{k=1}^K \hat w_k \log P^{\psi(\theta)}(\xi^k). \tag{3.8}$$

This objective (3.8) is the empirical projection of the intermediate trust-region target back to the parameterized twisted family. A more detailed derivation is given in Appendix B.

Algorithm Interpretation. At a high level, Algorithm 1 alternates between a twisted SMC sampling stage and a trust-region learning stage. At outer iteration i, Line 3 defines the current twisted

Algorithm 1 Trust-Region Iterative Twisted Sequential Monte Carlo (TRI-TSMC)

Require: Base model $\{\mu, f_t\}_{t=1}^T$, potentials $\{g_t\}_{t=0}^T$, trust-region size ϵ, number of particles K, outer iterations I.

1: Initialize twisting parameters θ_0.
2: **for** $i = 0, 1, \ldots, I-1$ **do**
3: Let $P_i = P^{\psi(\theta_i)}$ be the current twisted path measure.
4: Run the twisted SMC particle system under $\psi(\theta_i)$ to obtain ancestral trajectories $\{X_{0:T}^{(k)}\}_{k=1}^K$.
5: Compute residual trajectory weights $w^{(k)} = w^{\psi(\theta_i)}(X_{0:T}^{(k)})$ using (3.2).
6: Solve the empirical dual problem (3.7) to obtain $\hat{\lambda}_i$ and set $\hat{\tau}_i = 1/(1+\hat{\lambda}_i)$.
7: Calculate normalized tempered weights $\hat{w}_k = \frac{(w^{(k)})^{\hat{\tau}_i}}{\sum_{j=1}^K (w^{(j)})^{\hat{\tau}_i}}$, for $k = 1, \ldots, K$.
8: Update θ_{i+1} by optimizing from θ_i on $\mathcal{L}_i(\theta) = -\sum_{k=1}^K \hat{w}_k \log P^{\psi(\theta)}(X_{0:T}^{(k)})$.
9: **end for**
10: **return** θ_I.

path measure $P_i = P^{\psi(\theta_i)}$ induced by the current twisting function. The first stage (Line 4-5) generates particles (trajectories) using the current twisted SMC sampler. Particles are propagated by the twisted kernels in (3.1) and resampled using the residual incremental weights. At each step t,

$$W_t^k = w_t^{\psi(\theta_i)}(X_t^k)/\textstyle\sum_{\ell=1}^K w_t^{\psi(\theta_i)}(X_t^\ell), \qquad A_t^k \sim \mathrm{Cat}(W_t^{1:K}), \qquad X_{t+1}^k \sim f_{t+1}^{\psi(\theta_i)}(\cdot \mid X_t^{A_t^k}).$$

The resulting resampled trajectories form the empirical particle measure used for twist learning. Line 5 then computes the residual trajectory weights $w^{(k)}$, which measure the remaining mismatch between the current twisted proposal and the unnormalized target measure. The second stage (Lines 6-8) performs the trust-region update and projection. Line 6 solves the empirical dual problem to obtain $\hat{\lambda}_i$ and the trust-region temperature $\hat{\tau}_i = 1/(1+\hat{\lambda}_i)$. Line 7 uses this temperature to construct tempered trajectory weights. These weights define a trust-region-reweighted empirical distribution on the realized particle system. Finally, Line 8 fits the next twisting function by weighted maximum likelihood, so that $P^{\psi(\theta_{i+1})}$ approximates this intermediate target within the parameterized twisted family.

4 Theoretical Analysis

This section analyzes the theoretical guarantees of TRI-TSMC. We first formalize the value-function interpretation of the optimal twisting function and show that the optimal twisting function yields a zero-variance sampler. We demonstrate that the exact KL-constrained update follows an escort path from the current twisted proposal toward the target distribution. Finally, we prove that the weighted maximum-likelihood step is a forward-KL projection and that the trust-region path reduces residual importance-weight variance. Throughout this section, we assume that $g_t(x_t) > 0$ for all t and all x_t. We denote $P_i = P^{\psi(\theta_i)}$ and use R_i defined in (3.4) for path-space analysis. Complete proofs in this section are given in Appendix C.

Theorem 4.1 (Equivalence to soft value function)**.** Let $\psi^* = \{\psi_t^*\}_{t=0}^T$ be the optimal twisting function sequence satisfying (3.3). Define $V_t^*(x_t) := \alpha \log \psi_t^*(x_t)$ for $t = 0, \ldots, T$. Then, for $t = 0, \ldots, T-1$, V_t^* satisfies the soft Bellman recursion

$$V_t^*(x_t) = \alpha \log g_t(x_t) + \alpha \log \int \exp\left(V_{t+1}^*(x_{t+1})/\alpha\right) f_{t+1}(dx_{t+1} \mid x_t), \tag{4.1}$$

with terminal condition $V_T^*(x_T) = \alpha \log g_T(x_T)$. Crucially, under the optimal twist, $w^{\psi^*}(X_{0:T}) = Z$, which implies a zero-variance importance sampler.

Remark 4.2. In the terminal-reward case, where $g_t(x_t) \equiv 1$ for $t < T$ and $g_T(x_T) = \exp(r(x_T)/\alpha)$, Theorem 4.1 reduces to $V_t^*(x_t) = \alpha \log \int \exp(V_{t+1}^*(x_{t+1})/\alpha) f_{t+1}(dx_{t+1} \mid x_t)$ for $t = 0, \ldots, T-1$, with $V_T^*(x_T) = r(x_T)$. Thus, in diffusion alignment, the optimal twisting function can be viewed as the soft value function of the current denoising state.

We next analyze the exact path-space update used to move the current proposal toward the target. The following proposition shows that the exact update in Lemma 3.1 lies on an escort path between the current twisted proposal P_i and the target distribution π.

Proposition 4.3. Suppose that, for every $\tau \in [0,1]$, $\mathbb{E}_{X\sim P_i}[R_i(X)^\tau |\log R_i(X)|^2] < \infty$. Define

$$q_{i,\tau}(dX) := R_i(X)^\tau P_i(dX)/\mathbb{E}_{X\sim P_i}[R_i(X)^\tau], \tag{4.2}$$

$$\Lambda_i(\tau) := \log \mathbb{E}_{X\sim P_i}[R_i(X)^\tau]. \tag{4.3}$$

Then $q_{i,0} = P_i$, $q_{i,1} = \pi$, and the exact trust-region solution in (3.5) satisfies $q_i^* = q_{i,\tau_i}$ and $\tau_i = 1/(1+\lambda_i)$, where λ_i is the dual optimizer in Lemma 3.1. Moreover, $\tau \mapsto D_{\mathrm{KL}}(q_{i,\tau}\|P_i)$ is nondecreasing on $[0,1]$, while $\tau \mapsto D_{\mathrm{KL}}(q_{i,\tau}\|\pi)$ is nonincreasing on $[0,1]$. If $P_i \neq \pi$, then these monotonicities are strict on $(0,1]$ and $[0,1)$, respectively. Consequently, for every $\epsilon \in (0, D_{\mathrm{KL}}(\pi\|P_i))$, there exists a unique $\tau_i \in (0,1)$ such that $D_{\mathrm{KL}}(q_{i,\tau_i}\|P_i) = \epsilon$.

The escort update q_i^* is defined in the space of path measures and is generally not represented exactly by the parameterized twisted family. Therefore, the implementable algorithm must project this intermediate target back to a tractable family $P^{\psi(\theta)}$. The next lemma shows that the weighted maximum-likelihood step used by TRI-TSMC is precisely the forward-KL projection of q_i^*.

Lemma 4.4. Define the projection objective $L_i(\theta) := -\mathbb{E}_{X\sim q_i^*}[\log P^{\psi(\theta)}(X)]$. Then minimizing $L_i(\theta)$ over θ is equivalent to minimizing $D_{\mathrm{KL}}(q_i^*\|P^{\psi(\theta)})$. More precisely, $L_i(\theta) = D_{\mathrm{KL}}\big(q_i^*\|P^{\psi(\theta)}\big) - \mathbb{E}_{X\sim q_i^*}\big[\log q_i^*(X)\big]$. Consequently, if θ_{i+1} satisfies $L_i(\theta_{i+1}) \le \inf_\theta L_i(\theta) + \eta_i$, then, with $P_{i+1} := P^{\psi(\theta_{i+1})}$, we have $D_{\mathrm{KL}}\big(q_i^*\|P_{i+1}\big) \le \inf_\theta D_{\mathrm{KL}}\big(q_i^*\|P^{\psi(\theta)}\big) + \eta_i$.

Thus, the weighted maximum-likelihood objective in (3.8) is the empirical version of a forward-KL projection from the exact trust-region target to the learned twisted path measure. Appendix C further gives a projection error bound for this step. Finally, we analyze the stability of the tempered reweighting step. Since the projection objective is weighted by importance weights, a natural measure of stability is the variance of the residual weights after moving along the escort path.

Theorem 4.5 (Variance reduction)**.** Assume that $R_i \in L^2(P_i)$. Let $q_{i,\tau}$ and $\Lambda_i(\tau)$ be defined by (4.2) and (4.3), and define $\chi^2(\pi\|q) := \int (\frac{d\pi}{dq}(X) - 1)^2 q(dX)$. Then, for every $\tau \in [0,1]$,

$$Z^{-2}\mathrm{Var}_{X\sim q_{i,\tau}}\left[\frac{d\gamma}{dq_{i,\tau}}(X)\right] = \chi^2(\pi\|q_{i,\tau}). \tag{4.4}$$

Further, $\tau \mapsto \chi^2(\pi\|q_{i,\tau})$ is nonincreasing on $[0,1]$ and strictly decreasing on $[0,1)$ for $P_i \neq \pi$.

Remark 4.6. Theorem 4.5 gives a distribution-level explanation for the stability of the trust-region update. Along the exact escort path, it monotonically reduces the normalized variance of the residual importance weights. This suggests that, as the proposal moves toward the target, fewer trajectories dominate the weighted maximum-likelihood projection, leading to a more stable update.

5 Experiments

We now evaluate TRI-TSMC on two inference-time alignment tasks: discrete diffusion text generation and text-to-image diffusion generation. The first task studies whether TRI-TSMC can steer a Masked Diffusion Language Model toward fluent and grammatically acceptable text. The second task evaluates TRI-TSMC on text-to-image Stable Diffusion models toward images with higher ImageReward. We present the performance of TRI-TSMC across different tasks compared with Best-of-N and FK-Steering under comparable sampling budgets.

5.1 Discrete Diffusion Text Generation

Experimental Setup. We first evaluate TRI-TSMC on inference-time alignment for discrete diffusion text generation. We use the Masked Diffusion Language Model (MDLM) (Sahoo et al., 2024) trained on OpenWebText as the base generator. Our evaluation uses 15 discriminative prompts from the PPLM benchmark (Dathathri et al., 2019). For each prompt and each random seed in $\{42, 43, 44\}$, we generate 20 batches, yielding 300 samples per seed and 900 samples in total. All experiments use 200 diffusion steps, and particle resampling is performed every 20 steps.

For text generation experiments, we use a mixed steering reward that combines a PPL reward with a grammar reward CoLA: $r_{\text{mix}} = r_{\text{PPL}} + \beta r_{\text{CoLA}}$. Here r_{PPL} is the log negative perplexity score derived from a language-model perplexity evaluator, r_{CoLA} is the acceptability score from a CoLA classifier (Warstadt et al., 2019), and β is a fixed mixing weight. In the main experiment, we use GPT-2 (Radford et al., 2019) as reward function and provide Qwen2.5-1.5B (Hui et al., 2024) PPL-reward-based results in Appendix D.3. We evaluate PPL with GPT-2 in all text experiments and also report CoLA and Dist-1/2/3 (Li et al., 2016).

Baselines. We compare TRI-TSMC against three baselines: (i) **Base**, which samples directly from the pretrained MDLM without steering ($K = 1$); (ii) **Best-of-N (BoN)**, which draws $K \in \{16, 48\}$ independent samples and returns the one with the highest final reward; (iii) **FK-Steering**, the particle-based inference-time steering method of Singhal et al. (2025) using the reward r_{PPL} to define potential functions for alignment, with $K \in \{16, 48\}$. The $K = 16$ setting matches the number of particles used by TRI-TSMC in each sampling iteration, while $K = 48$ matches the total number of sampled trajectories used by TRI-TSMC across all iterations.

Implementation Details. We use the same resampling interval for FK-Steering and TRI-TSMC, performing particle resampling every 20 reverse diffusion steps. For GPT-2-based alignment, TRI-TSMC uses the `max` potential to construct local Feynman–Kac weights from intermediate reward estimation. The twisting function is parameterized by an MLP. Full hyperparameters, reward weights, potential definitions, and additional implementation details are provided in Appendix D.2.

Results Analysis. Table 1 reports GPT-2-based PPL reward alignment results on MDLM. TRI-TSMC achieves the lowest perplexity among all inference-time baselines, reducing PPL from 36.702 under the strongest FK-Steering baseline ($K = 48$) and 44.109 under BoN ($K = 48$) to 29.781 with $K = 16$ particles. TRI-TSMC also attains the highest CoLA score, improving over FK-Steering ($K = 48$) from 0.393 to 0.727. In terms of diversity, TRI-TSMC has lower Dist-1/2/3 scores than FK-Steering, decreasing from $0.520/0.831/0.888$ under FK-Steering ($K = 48$) to $0.488/0.774/0.833$. The Base model still has the highest Dist-n values overall, which is expected because it performs no reward alignment and therefore samples more broadly. Overall, these results show that TRI-TSMC gives the strongest improvement on the primary alignment objective and substantially improves CoLA, while trading off some diversity. Note that TRI-TSMC uses $K = 16$ particles for each outer iteration and runs 3 outer iterations, giving the same total trajectory budget as FK-Steering with $K = 48$ while using a smaller particle set per iteration.

Table 1: Alignment results on MDLM under GPT-2-based evaluation. PPL is the primary alignment objective, while CoLA and Dist-n are secondary quality indicators. Lower PPL is better; higher CoLA and Dist-n are better.

Method	K	PPL(*) ↓	CoLA ↑	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑
Base	1	104.427	0.223	**0.594**	**0.919**	**0.941**
BoN	16	51.301	0.347	0.552	0.879	0.922
BoN	48	44.109	0.371	0.532	0.852	0.904
FK-Steering	16	43.050	0.390	0.528	0.849	0.905
FK-Steering	48	36.702	0.393	0.520	0.831	0.888
TRI-TSMC	16	**29.781**	**0.727**	0.488	0.774	0.833

We further provide Qwen2.5-1.5B-based evaluation in Appendix D.3. Similarly, TRI-TSMC achieves the lowest PPL and highest CoLA score among the methods. In addition, we conduct an FK-Steering mixed-reward ablation study to further study the gain of mixed reward design in Appendix D.4. Finally, we further provide a qualitative analysis of generated text samples that illustrate the same trade-off between fluency, acceptability, and diversity in Appendix D.9.

5.2 Text-to-Image Diffusion Models

Experimental setup. We further evaluate TRI-TSMC on text-to-image inference-time alignment. We leverage the Stable Diffusion (Rombach et al., 2022; Podell et al., 2023) as the base model and sample images conditioned on a text prompt. We use Stable Diffusion v1.5 (SD v1.5) as the main base generator and report additional results on SDXL to show the generalizability of TRI-TSMC. The reward used for steering is the off-the-shelf ImageReward (IR) model (Xu et al., 2023), which serves as the primary alignment objective in this experiment. We also report GenEval (Ghosh et al., 2023) for prompt fidelity and HPS (Wu et al., 2023b) as an additional human preference metric. For evaluating the IR and HPS score, we use the prompts from the ImageReward benchmark, which contains 100 prompts to evaluate alignment performance. We also use the official GenEval

benchmark which contains 553 prompts to evaluate compositional prompt fidelity. We choose the same three random seeds as the text generation task with $\{42, 43, 44\}$ to report the results.

Baselines. We compare TRI-TSMC against four baselines: (i) **Base**, which samples directly from the diffusion model without reward steering ($K = 1$); (ii) **Best-of-N (BoN)**, which draws $K \in \{4, 8\}$ independent images and returns the one with the highest IR score; (iii) **FK-Steering**, the particle-based inference-time steering method of Singhal et al. (2025) that uses IR-based Feynman–Kac potentials for particle weighting and resampling, with $K \in \{4, 8\}$; (iv) **Post-training baselines**, including Diffusion-DPO (Wallace et al., 2024). For BoN and FK-Steering, we report both $K = 4$ and $K = 8$ results. The $K = 4$ setting matches the number of particles used by TRI-TSMC in one iteration, while the $K = 8$ setting matches the total number of sampled trajectories used by TRI-TSMC across all iterations.

Implementation Details. For all image-generation experiments, we use the base diffusion model as the proposal generator and adopt DDIM sampling with 100 denoising steps and classifier-free guidance scale 7.5, consistent with the setup used by Singhal et al. (2025). For particle-based inference-time methods, we use a resampling frequency of 20, corresponding to the resampling schedule $[20, 40, 60, 80, 99]$. At each resampling stage, we compute an intermediate reward on the denoised prediction $\hat{x}_0$, which provides a practical estimate of final image quality before the terminal denoising step. For SD v1.5 tasks, TRI-TSMC use the `diff` potential to construct local Feynman–Kac weights. The twisting function is parameterized by a CNN. We use $K = 4$ particles for all experiments in text-to-image tasks. More details can be found in Appendix E.1.

Results analysis. Table 2 reports the main text-to-image results on SD v1.5. IR is the primary objective in this experiment. TRI-TSMC achieves the best IR score among all baselines. It improves over Best-of-N ($K = 8$) from 0.945 to 0.979, and over FK-Steering ($K = 8$) from 0.930 to 0.979. The improvement is larger when compared with the $K = 4$ baselines. Overall, these results show that TRI-TSMC steers the sampling process toward images preferred by IR and achieves the best score among all the baselines with comparable sample trajectories. On the secondary metrics, TRI-TSMC obtains an HPS score of 0.270. This is comparable to FK-Steering ($K = 8$), but lower than Best-of-N ($K = 8$). This is reasonable because TRI-TSMC is trained and selected using IR rather than HPS. For GenEval, TRI-TSMC is comparable to FK-Steering in the current evaluation. This suggests that the improvement in IR does not come with a large drop in prompt fidelity.

Table 2: Main results on text-to-image alignment with SD v1.5. IR is the primary alignment metric. GenEval evaluates compositional prompt fidelity, and HPS evaluates human preference.

Method	K	IR(*) ↑	GenEval ↑	HPS ↑
Base	1	0.233	0.428	0.246
DPO	1	0.359	0.441	0.256
Best-of-N	4	0.750	0.599	0.267
Best-of-N	8	0.945	**0.670**	**0.276**
FK-Steering	4	0.718	0.553	0.261
FK-Steering	8	0.930	0.630	0.270
TRI-TSMC	4	**0.979**	0.601	0.270

On SDXL, TRI-TSMC again achieves the highest IR, with HPS comparable to the baselines (Table 13). Figure 3 compares particles before and after one trust-region update. The updated particles are more consistently steered toward generations preferred by IR than at initialization.



Figure 3: Effect of a trust-region twist update for the prompt "footage of an astronaut." The left four images show initialization particles, and the right four show particles after one iteration.

6 Conclusion

We introduced TRI-TSMC for inference-time alignment of diffusion models. The method alternates a KL-constrained path-space update with weighted maximum-likelihood fitting of the twisting function. The optimal twisting function has a soft-value interpretation and yields a zero-variance importance sampler. The exact trust-region update follows an escort path that decreases residual importance-weight variance. On MDLM text generation, TRI-TSMC improves PPL and CoLA, and on Stable Diffusion it improves ImageReward, under matched total sampled-trajectory budgets. Future work will address trade-offs in text diversity and image compositional fidelity.

AI USE STATEMENT

In this work, we used generative AI tools only for coding and polishing the writing. We have not used generative AI tools for research ideation, methodology or experiment design, theoretical development or proofs, data generation or analysis, or other tasks with required disclosure. We have reviewed all AI-assisted work. AI-assisted code was verified and tested for correctness, and AI-polished writing was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the source code in the supplementary materials. The main paper describes the proposed method, algorithm, experimental setups, evaluation protocols, and the models and benchmarks used in our experiments. Additional implementation details, model architectures, hyperparameters, and experimental configurations are provided in the appendix. For the theoretical results, we state the required assumptions and provide complete proofs in the appendix.

REFERENCES

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Denis Blessing, Julius Berner, Lorenz Richter, Carles Domingo-Enrich, Yuanqi Du, Arash Vahdat, and Gerhard Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference. *arXiv preprint arXiv:2508.12511*, 2025.

Nicola Branchini and Víctor Elvira. Optimized auxiliary particle filters: adapting mixture proposals via convex optimization. In *Uncertainty in Artificial Intelligence*, pp. 1289–1299. PMLR, 2021.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Riccardo Barbano, Vincent Dutordoir, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. Deft: Efficient fine-tuning of diffusion models by learning the generalised h-transform. *Advances in Neural Information Processing Systems*, 37:19636–19682, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

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.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 110–119, 2016.

Jianfeng Lu and Yuliang Wang. Guidance for twisted particle filter: a continuous-time perspective, 2024. URL https://arxiv.org/abs/2409.02399.

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.

Chinmay Pani, Zijing Ou, and Yingzhen Li. Test-time alignment of discrete diffusion models with sequential monte carlo. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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.

Oswin So, Brian Karrer, Chuchu Fan, Ricky TQ Chen, and Guan-Horng Liu. Discrete adjoint matching. *arXiv preprint arXiv:2602.07132*, 2026.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

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.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023a.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023b.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

A Related Work

Diffusion Model Alignment Diffusion alignment studies steering a pretrained diffusion model toward a reward-tilted target distribution of the form $p^{\text{target}}(x) \propto p^{\text{pre}}(x)\exp(r(x)/\alpha)$, where $r(x)$ is a task-specific reward, possibly provided by a black-box evaluator or a learned reward model, and $\alpha > 0$ is a temperature parameter. Existing methods can be broadly grouped into three directions: post-training, inference-time alignment, and hybrid distillation.

Post-training methods directly update model parameters so that the learned reverse process shifts toward high-reward samples while staying close to the pretrained model. Black et al. (2024) formulates diffusion denoising as a sequential decision process and applies policy-gradient fine-tuning. Other methods directly backpropagate through the denoising chain when the reward is differentiable (Clark et al., 2023), or introduce preference-based objectives and KL regularization to improve stability and reduce drift from the base model (Fan et al., 2023; Wallace et al., 2024). These methods can produce efficient aligned samplers after training, but they require updating model weights and are usually tied to a specific reward.

Inference-time alignment keeps the pretrained model fixed and modifies the sampling procedure instead. This line of work is attractive when rewards are non-differentiable, expensive to evaluate, or frequently changed across tasks. A classical direction is gradient-based guidance, where each reverse step is adjusted using gradients from an external signal. Classifier guidance (Dhariwal & Nichol, 2021) is the standard example, and later work extends this idea to more general posterior sampling and inverse problems (Chung et al., 2022). Another direction uses particle-based methods such as Sequential Monte Carlo to approximate reward-tilted target distributions. These methods are especially useful when guidance is only available through a terminal reward or a black-box evaluator, so there is no simple gradient signal at each step. Recent work has developed several SMC-based approaches for inference-time alignment in continuous diffusion models and discrete diffusion language models. FK-Steering formulates reward-guided diffusion generation as a Feynman–Kac particle system and shows that increasing the inference-time particle budget can improve alignment quality without updating the base model (Singhal et al., 2025). Twisted Diffusion Sampler (TDS) similarly uses SMC for diffusion models, with a focus on practical and asymptotically exact conditional sampling (Wu et al., 2023a). Beyond standard particle steering, optimized auxiliary particle filters study how to adapt proposal mixtures through convex optimization (Branchini & Elvira, 2021), while test-time SMC alignment for discrete diffusion constructs training-free local proposals using Taylor and Gumbel–Softmax approximations (Pani et al., 2025). For masked diffusion language models, ILRR performs reference-based representation steering (Avrahami & Nachmani, 2026), whereas Self-Rewarding SMC uses trajectory-level model confidence as an intrinsic signal for particle reweighting and resampling (Luo et al., 2026).

A third direction combines inference-time alignment and post-training through distillation. The main idea is to first run a stronger inference-time method as a teacher, and then train a student model to imitate the guided trajectories or conditional transitions. This can convert a slow but accurate inference-time sampler into a fast aligned generator. Progressive distillation was originally proposed for acceleration (Salimans & Ho, 2022), and recent work discusses policy distillation more broadly as a way to transfer guidance, SMC, or value-based steering back into model weights (Uehara et al., 2025). Related to these directions, several recent papers study diffusion alignment from an optimal-control perspective and connect aligned sampling to Doob's h-transform or stochastic control formulations (Domingo-Enrich et al., 2024; Denker et al., 2024; So et al., 2026). This view is closely related to the value-function interpretation of twisting functions used in our method.

B Further Details in Preliminary and Methodology

This section provides additional details for Section 2 and Section 3.

Standard SMC sampling. Consider the Feynman–Kac target

$$\gamma(dx_{0:T}) = P(dx_{0:T})\prod_{t=0}^{T} g_t(x_t), \qquad P(dx_{0:T}) = \mu(dx_0)\prod_{t=1}^{T} f_t(dx_t \mid x_{t-1}).$$

Standard SMC keeps a set of particles and their ancestral paths. We initialize $X_0^k \sim \mu$ for $k = 1, \dots, K$. At time t, the current particles are weighted by the local potential,

$$W_t^k = \frac{g_t(X_t^k)}{\sum_{\ell=1}^K g_t(X_t^\ell)}, \qquad k = 1, \dots, K.$$

We then sample ancestors $A_t^k \sim \mathrm{Cat}(W_t^{1:K})$ and propagate

$$X_{t+1}^k \sim f_{t+1}(\cdot \mid X_t^{A_t^k}), \qquad k = 1, \dots, K.$$

The ancestral path is copied together with the ancestor: if $A_t^k = a$, then the new path of particle k first inherits $X_{0:t}^a$ and then appends the newly sampled state X_{t+1}^k. Thus, high-weight particles are duplicated and low-weight particles are removed before the next propagation step. In practice, resampling does not need to occur at every step; if t is not a resampling stage, we simply set $A_t^k = k$ and propagate each particle forward without resampling.

This procedure approximates the Feynman–Kac path measure by repeatedly correcting the base dynamics with the potentials g_t. However, if the base dynamics f_t is far from the reward-tilted target, many particles may still move through low-reward regions before being corrected by weighting and resampling. Twisted SMC modifies this propagation step directly.

Twisted SMC sampling. Given a positive twisting function sequence $\psi = \{\psi_t\}_{t=0}^T$, twisted SMC uses the twisted initial distribution and transition kernels

$$\mu^\psi(dx_0) = \frac{\psi_0(x_0)}{c_\psi}\mu(dx_0), \qquad f_t^\psi(dx_t \mid x_{t-1}) = \frac{\psi_t(x_t)}{\tilde{\psi}_{t-1}(x_{t-1})} f_t(dx_t \mid x_{t-1}),$$

where $c_\psi = \int \psi_0(x_0)\mu(dx_0)$ and $\tilde{\psi}_{t-1}(x_{t-1}) = \int \psi_t(x_t) f_t(dx_t \mid x_{t-1})$. The role of ψ_t is to give look-ahead guidance: states with larger estimated future potential are assigned larger probability under the twisted transition.

After twisting the transition, the original potentials are replaced by residual incremental weights. With the convention $\tilde{\psi}_T(\cdot) \equiv 1$, these residual weights are

$$w_0^\psi(x_0) = g_0(x_0) c_\psi \frac{\tilde{\psi}_0(x_0)}{\psi_0(x_0)}, \qquad w_t^\psi(x_t) = g_t(x_t)\frac{\tilde{\psi}_t(x_t)}{\psi_t(x_t)}, \quad t = 1, \dots, T.$$

Their product recovers the global residual importance weight

$$w^\psi(X_{0:T}) = w_0^\psi(X_0) \prod_{t=1}^T w_t^\psi(X_t) = \frac{d\gamma}{dP^\psi}(X_{0:T}).$$

Therefore, twisting changes the proposal but preserves the target through the residual weights.

The twisted SMC sampler is the same resample–propagate procedure as standard SMC, except that both the propagation kernel and the weights are changed. We initialize $X_0^k \sim \mu^\psi$. At a resampling stage t, we compute

$$W_t^k = \frac{w_t^\psi(X_t^k)}{\sum_{\ell=1}^K w_t^\psi(X_t^\ell)}, \qquad k = 1, \dots, K.$$

We then sample ancestors $A_t^k \sim \mathrm{Cat}(W_t^{1:K})$ and propagate with the twisted kernel,

$$X_{t+1}^k \sim f_{t+1}^\psi(\cdot \mid X_t^{A_t^k}), \qquad k = 1, \dots, K.$$

As in standard SMC, the ancestral path of particle k is updated by copying the path of its sampled ancestor and appending the new state. If no resampling is performed at step t, we set $A_t^k = k$ and only apply the twisted transition.

This makes the role of twisting functions explicit. Standard SMC uses the base kernel f_t and relies on the potentials g_t to correct particles after propagation. Twisted SMC instead uses ψ_t to modify the transition before propagation and then uses the residual weights w_t^ψ to correct the remaining mismatch. A good twisting function therefore reduces the variance of the residual weights. In the ideal case, the optimal twisting function makes these residual weights constant, giving a zero-variance importance sampler.

Incremental weights for twisted SMC. Using (3.2), the residual importance weight under the twisted proposal can be decomposed as

$$w^{\psi}(X_{0:T}) = w_0^{\psi}(X_0) \prod_{t=1}^{T} w_t^{\psi}(X_t),$$

where

$$w_0^{\psi}(x_0) = g_0(x_0) c_{\psi} \frac{\tilde{\psi}_0(x_0)}{\psi_0(x_0)}, \qquad w_t^{\psi}(x_t) = g_t(x_t) \frac{\tilde{\psi}_t(x_t)}{\psi_t(x_t)}, \quad t = 1, \dots, T.$$

Here we use the convention $\tilde{\psi}_T(\cdot) \equiv 1$. If $\psi = \psi^*$, then $\psi_t^*(x_t) = g_t(x_t)\tilde{\psi}_t^*(x_t)$ for $t < T$ and $\psi_T^*(x_T) = g_T(x_T)$. Therefore $w_t^{\psi^*}(x_t) \equiv 1$ for $t = 1, \dots, T$. For the initial term, $w_0^{\psi^*}(x_0) = c_{\psi^*}$, and applying the recursion in (3.3) gives

$$c_{\psi^*} = \int \mu(dx_0)\psi_0^*(x_0) = \int P(dx_{0:T}) \prod_{t=0}^{T} g_t(x_t) = Z.$$

Thus $w^{\psi^*}(X_{0:T}) = Z$, which gives the zero-variance importance sampler.

Geometric annealing interpretation. The trust-region update in Lemma 3.1 gives

$$\frac{dq_i^*}{dP_i}(X_{0:T}) \propto \left(\frac{d\pi}{dP_i}(X_{0:T}) \right)^{\frac{1}{1+\lambda_i}}.$$

If the update is applied exactly in path space, i.e., $P_{i+1} = q_i^*$, then

$$\frac{dP_{i+1}}{dP_i}(X_{0:T}) \propto \left(\frac{d\pi}{dP_i}(X_{0:T}) \right)^{\frac{1}{1+\lambda_i}}.$$

Assume that P_i lies on a geometric path between P_0 and π, namely

$$\frac{dP_i}{dP_0}(X_{0:T}) \propto \left(\frac{d\pi}{dP_0}(X_{0:T}) \right)^{\beta_i}.$$

Then the next exact update satisfies the same form with

$$\beta_{i+1} = \beta_i + (1 - \beta_i) \frac{1}{1 + \lambda_i}.$$

Equivalently, starting from $\beta_0 = 0$,

$$\beta_{i+1} = 1 - \prod_{m=0}^{i} \frac{\lambda_m}{1 + \lambda_m}.$$

This gives an annealing interpretation of the exact trust-region path: each update moves the proposal from the current twisted path measure toward the target path measure by a controlled amount.

Dual objective. The dual formulation corresponding to (3.5) is

$$\lambda_i = \operatorname*{argmin}_{\lambda \geq 0} \left[\lambda\epsilon + (1 + \lambda) \log \mathbb{E}_{X \sim P_i} \left[\left(R_i(X) \right)^{\frac{1}{1+\lambda}} \right] \right].$$

Since $R_i(X) = Z^{-1} w^{\psi(\theta_i)}(X)$, replacing R_i by the unnormalized weight $w^{\psi(\theta_i)}$ only changes the objective by an additive constant independent of λ. Therefore, the empirical dual in (3.7) can be computed directly using unnormalized importance weights.

Weighted maximum-likelihood projection. The trust-region step produces an ideal intermediate distribution q_i^* in path space. Projecting it back to the parameterized twisted family gives

$$\theta_{i+1} = \operatorname*{argmin}_\theta D_{\mathrm{KL}}\big(q_i^* \| P^{\psi(\theta)}\big) = \operatorname*{argmax}_\theta \mathbb{E}_{X\sim q_i^*}\big[\log P^{\psi(\theta)}(X)\big].$$

Using importance reweighting with respect to P_i, this can be written as

$$\theta_{i+1} = \operatorname*{argmax}_\theta \mathbb{E}_{X\sim P_i}\left[\frac{dq_i^*}{dP_i}(X)\log P^{\psi(\theta)}(X)\right].$$

By Lemma 3.1,

$$\frac{dq_i^*}{dP_i}(X) \propto \Big(R_i(X)\Big)^{\frac{1}{1+\lambda_i}} \propto \Big(w^{\psi(\theta_i)}(X)\Big)^{\frac{1}{1+\lambda_i}}.$$

Replacing the expectation by sampled trajectories gives the weighted maximum-likelihood objective in (3.8).

Finally, the trajectory log-probability under a twisted model decomposes as

$$\log P^{\psi(\theta)}(\xi^k) = \log \mu^{\psi(\theta)}(\xi_0^k) + \sum_{t=1}^{T} \log f_t^{\psi(\theta)}(\xi_t^k \mid \xi_{t-1}^k).$$

Equivalently, using the definition of the twisted kernels in (3.1),

$$\begin{aligned}\log P^{\psi(\theta)}(\xi^k) = {} & \log \mu(\xi_0^k) + \log \psi_0(\xi_0^k;\theta) - \log c_{\psi(\theta)} \\ & + \sum_{t=1}^{T}\Big[\log f_t(\xi_t^k \mid \xi_{t-1}^k) + \log \psi_t(\xi_t^k;\theta) - \log \tilde{\psi}_{t-1}(\xi_{t-1}^k;\theta)\Big].\end{aligned}$$

This is the quantity optimized in the weighted maximum-likelihood projection. In discrete diffusion language models, the normalizers $\tilde{\psi}_{t-1}$ are typically tractable because they are sums over the discrete state space.

C Proofs of Theoretical Results

C.1 Proof of Lemma 3.1

Proof. Consider the Lagrangian

$$\mathcal{L}(q,\lambda) = D_{\mathrm{KL}}(q\|\pi) + \lambda\Big(D_{\mathrm{KL}}(q\|P_i) - \epsilon\Big),$$

with $\lambda \geq 0$. Optimizing $\mathcal{L}(q,\lambda)$ over distributions q yields the Gibbs-form solution

$$\begin{aligned} q_i^*(X_{0:T}) &\propto \pi(X_{0:T})^{\frac{1}{1+\lambda_i}} P_i(X_{0:T})^{\frac{\lambda_i}{1+\lambda_i}} \\ &= P_i(X_{0:T})\left(\frac{\pi(X_{0:T})}{P_i(X_{0:T})}\right)^{\frac{1}{1+\lambda_i}} \\ &= P_i(X_{0:T})\Big(R_i(X_{0:T})\Big)^{\frac{1}{1+\lambda_i}}, \end{aligned}$$

which gives (3.6). Since P_i is feasible for (3.5) because $D_{\mathrm{KL}}(P_i\|P_i) = 0 \leq \epsilon$, optimality of q_i^* implies

$$D_{\mathrm{KL}}(q_i^*\|\pi) \leq D_{\mathrm{KL}}(P_i\|\pi).$$

Finally, if $P_{i+1} = q_i^*$ exactly, then

$$\frac{dP_{i+1}}{dP_i}(X_{0:T}) \propto \Big(R_i(X_{0:T})\Big)^{\frac{1}{1+\lambda_i}} = \left(\frac{d\pi}{dP_i}(X_{0:T})\right)^{\frac{1}{1+\lambda_i}}.$$

Expanding this recursion gives the geometric annealing path

$$\frac{dP_i}{dP_0}(X_{0:T}) \propto \left(\frac{d\pi}{dP_0}(X_{0:T})\right)^{\beta_i}, \qquad \beta_{i+1} = \beta_i + (1-\beta_i)\frac{1}{1+\lambda_i}.$$

Equivalently, $\beta_i = 1 - \prod_{m=0}^{i-1}\frac{\lambda_m}{1+\lambda_m}$. □

C.2 Proof of Theorem 4.1

Proof. By (3.3), the optimal twisting function satisfies

$$\psi_t^*(x_t) = g_t(x_t) \int \psi_{t+1}^*(x_{t+1}) f_{t+1}(dx_{t+1} \mid x_t), \qquad t = 0, \dots, T-1,$$

with terminal condition $\psi_T^*(x_T) = g_T(x_T)$. Taking logarithms on both sides and multiplying by α, we obtain

$$\alpha \log \psi_t^*(x_t) = \alpha \log g_t(x_t) + \alpha \log \int \psi_{t+1}^*(x_{t+1}) f_{t+1}(dx_{t+1} \mid x_t).$$

Using the definition $V_t^*(x_t) = \alpha \log \psi_t^*(x_t)$ in Theorem 4.1, this becomes

$$V_t^*(x_t) = \alpha \log g_t(x_t) + \alpha \log \int \exp\left(\frac{V_{t+1}^*(x_{t+1})}{\alpha}\right) f_{t+1}(dx_{t+1} \mid x_t),$$

which proves (4.1). The terminal condition follows directly from

$$V_T^*(x_T) = \alpha \log \psi_T^*(x_T) = \alpha \log g_T(x_T).$$

Next, for $t = 1, \dots, T$, the incremental twisted weights are

$$w_t^{\psi^*}(x_t) = g_t(x_t) \frac{\tilde{\psi}_t^*(x_t)}{\psi_t^*(x_t)}.$$

By (3.3), we have $\psi_t^*(x_t) = g_t(x_t)\tilde{\psi}_t^*(x_t)$, hence

$$w_t^{\psi^*}(x_t) = g_t(x_t) \frac{\tilde{\psi}_t^*(x_t)}{g_t(x_t)\tilde{\psi}_t^*(x_t)} = 1.$$

For $t = 0$, we have

$$w_0^{\psi^*}(x_0) = g_0(x_0) c_{\psi^*} \frac{\tilde{\psi}_0^*(x_0)}{\psi_0^*(x_0)}.$$

Using $\psi_0^*(x_0) = g_0(x_0)\tilde{\psi}_0^*(x_0)$ again gives $w_0^{\psi^*}(x_0) = c_{\psi^*}$, so it remains to show that $c_{\psi^*} = Z$. By definition,

$$c_{\psi^*} = \int \mu(dx_0) \psi_0^*(x_0).$$

Applying the recursion in (3.3) repeatedly, we obtain

$$c_{\psi^*} = \int \mu(dx_0) \prod_{t=1}^{T} f_t(dx_t \mid x_{t-1}) \prod_{t=0}^{T} g_t(x_t) = \int P(dx_{0:T}) \prod_{t=0}^{T} g_t(x_t) = Z,$$

Therefore $w^{\psi^*}(X_{0:T}) = Z$, which implies that the importance sampler has zero variance. □

C.3 Proof of Proposition 4.3

Proof. First, by definition,

$$q_{i,0}(dX) = \frac{1}{\mathbb{E}_{X \sim P_i}[1]} P_i(dX) = P_i(dX).$$

Since $\mathbb{E}_{X \sim P_i}[R_i(X)] = 1$, we also have

$$q_{i,1}(dX) = R_i(X) P_i(dX) = \pi(dX).$$

By Lemma 3.1,

$$\frac{dq_i^*}{dP_i}(X) \propto \Big(R_i(X)\Big)^{\frac{1}{1+\lambda_i}}.$$

After normalization, this is exactly the escort family (4.2) with $\tau_i = \frac{1}{1+\lambda_i}$.

Next, from (4.2) and (4.3),

$$\frac{dq_{i,\tau}}{dP_i}(X) = \exp\Big(\tau \log R_i(X) - \Lambda_i(\tau)\Big).$$

Hence we have

$$D_{\mathrm{KL}}(q_{i,\tau}\|P_i) = \tau \mathbb{E}_{X\sim q_{i,\tau}}[\log R_i(X)] - \Lambda_i(\tau),$$
$$D_{\mathrm{KL}}(q_{i,\tau}\|\pi) = (\tau-1)\mathbb{E}_{X\sim q_{i,\tau}}[\log R_i(X)] - \Lambda_i(\tau).$$

Differentiating (4.3), we obtain

$$\Lambda_i'(\tau) = \frac{\mathbb{E}_{X\sim P_i}[R_i(X)^\tau \log R_i(X)]}{\mathbb{E}_{X\sim P_i}[R_i(X)^\tau]} = \mathbb{E}_{X\sim q_{i,\tau}}[\log R_i(X)],$$
$$\Lambda_i''(\tau) = \mathrm{Var}_{X\sim q_{i,\tau}}\big(\log R_i(X)\big).$$

Therefore,

$$\frac{d}{d\tau} D_{\mathrm{KL}}(q_{i,\tau}\|P_i) = \tau\, \mathrm{Var}_{X\sim q_{i,\tau}}\big(\log R_i(X)\big) \geq 0,$$
$$\frac{d}{d\tau} D_{\mathrm{KL}}(q_{i,\tau}\|\pi) = -(1-\tau)\, \mathrm{Var}_{X\sim q_{i,\tau}}\big(\log R_i(X)\big) \leq 0.$$

This proves the monotonicity properties.

If $P_i \neq \pi$, then R_i is not almost surely constant under P_i. Since $q_{i,\tau} \ll P_i$ for every $\tau \in (0,1)$, this implies $\mathrm{Var}_{X\sim q_{i,\tau}}\big(\log R_i(X)\big) > 0$ for every $\tau \in (0,1)$. Hence $\tau \mapsto D_{\mathrm{KL}}(q_{i,\tau}\|P_i)$ is strictly increasing on $(0,1]$, and $\tau \mapsto D_{\mathrm{KL}}(q_{i,\tau}\|\pi)$ is strictly decreasing on $[0,1)$. Finally, since

$$D_{\mathrm{KL}}(q_{i,0}\|P_i) = 0, \qquad D_{\mathrm{KL}}(q_{i,1}\|P_i) = D_{\mathrm{KL}}(\pi\|P_i),$$

continuity and strict monotonicity imply that, for every $\epsilon \in (0, D_{\mathrm{KL}}(\pi\|P_i))$, there exists a unique $\tau_i \in (0,1)$ such that $D_{\mathrm{KL}}(q_{i,\tau_i}\|P_i) = \epsilon$. □

C.4 Proof of Lemma 4.4

Proof. By the definition of KL divergence,

$$D_{\mathrm{KL}}\big(q_i^*\|P^{\psi(\theta)}\big) = \mathbb{E}_{X\sim q_i^*}\left[\log \frac{q_i^*(X)}{P^{\psi(\theta)}(X)}\right] = \mathbb{E}_{X\sim q_i^*}[\log q_i^*(X)] - \mathbb{E}_{X\sim q_i^*}[\log P^{\psi(\theta)}(X)].$$

Rearranging gives

$$L_i(\theta) = D_{\mathrm{KL}}\big(q_i^*\|P^{\psi(\theta)}\big) - \mathbb{E}_{X\sim q_i^*}\big[\log q_i^*(X)\big].$$

Since the second term does not depend on θ, minimizing $L_i(\theta)$ is equivalent to minimizing $D_{\mathrm{KL}}(q_i^*\|P^{\psi(\theta)})$. Therefore, if $L_i(\theta_{i+1}) \leq \inf_\theta L_i(\theta) + \eta_i$, subtracting the same constant $-\mathbb{E}_{X\sim q_i^*}[\log q_i^*(X)]$ from both sides yields

$$D_{\mathrm{KL}}\big(q_i^*\|P_{i+1}\big) \leq \inf_\theta D_{\mathrm{KL}}\big(q_i^*\|P^{\psi(\theta)}\big) + \eta_i,$$

which proves the result. □

C.5 Projection Error Bound

The following result quantifies how much of the exact trust-region improvement remains after projecting q_i^* back to the parameterized twisted family.

Theorem C.1 (Projection error bound)**.** Let q_i^* be the exact trust-region update, and let $P_{i+1} = P^{\psi(\theta_{i+1})}$ be the projected twisted path measure obtained by the weighted maximum likelihood step. Define $\Delta_i := D_{\mathrm{KL}}(P_i\|\pi) - D_{\mathrm{KL}}(q_i^*\|\pi) \geq 0$ and $\delta_i^{\rightarrow} := D_{\mathrm{KL}}(q_i^*\|P_{i+1})$, $\delta_i^{\leftarrow} := D_{\mathrm{KL}}(P_{i+1}\|q_i^*)$. Assume that the log-density ratio $\ell_i(X) := \log \frac{dq_i^*}{d\pi}(X)$ is bounded, with $\|\ell_i\|_\infty \leq M_i$. Then

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) \leq D_{\mathrm{KL}}(P_i\|\pi) - \Delta_i + \delta_i^{\leftarrow} + M_i\sqrt{2\delta_i^{\rightarrow}}. \tag{C.1}$$

Proof. By definition, we have

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) = \mathbb{E}_{X\sim P_{i+1}}\left[\log \frac{dP_{i+1}}{d\pi}(X)\right].$$

Insert q_i^* between P_{i+1} and π:

$$\log \frac{dP_{i+1}}{d\pi}(X) = \log \frac{dP_{i+1}}{dq_i^*}(X) + \log \frac{dq_i^*}{d\pi}(X).$$

Hence we have

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) = D_{\mathrm{KL}}(P_{i+1}\|q_i^*) + \mathbb{E}_{X\sim P_{i+1}}[\ell_i(X)].$$

Adding and subtracting the expectation under q_i^* gives

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) = \delta_i^{\leftarrow} + \mathbb{E}_{X\sim q_i^*}[\ell_i(X)] + \Big(\mathbb{E}_{X\sim P_{i+1}}[\ell_i(X)] - \mathbb{E}_{X\sim q_i^*}[\ell_i(X)]\Big).$$

Since $\ell_i(X) = \log \frac{dq_i^*}{d\pi}(X)$, we have

$$\mathbb{E}_{X\sim q_i^*}[\ell_i(X)] = D_{\mathrm{KL}}(q_i^*\|\pi).$$

Therefore,

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) = \delta_i^{\leftarrow} + D_{\mathrm{KL}}(q_i^*\|\pi) + \Big(\mathbb{E}_{X\sim P_{i+1}}[\ell_i(X)] - \mathbb{E}_{X\sim q_i^*}[\ell_i(X)]\Big).$$

We now bound the last term. By the total variation inequality,

$$\Big|\mathbb{E}_{X\sim P_{i+1}}[\ell_i(X)] - \mathbb{E}_{X\sim q_i^*}[\ell_i(X)]\Big| \le 2\|\ell_i\|_\infty \,\mathrm{TV}(P_{i+1}, q_i^*).$$

Using $\|\ell_i\|_\infty \le M_i$ and Pinsker's inequality,

$$\mathrm{TV}(P_{i+1}, q_i^*) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(q_i^*\|P_{i+1})} = \sqrt{\frac{1}{2}\delta_i^{\rightarrow}},$$

we obtain

$$\mathbb{E}_{X\sim P_{i+1}}[\ell_i(X)] - \mathbb{E}_{X\sim q_i^*}[\ell_i(X)] \le M_i\sqrt{2\delta_i^{\rightarrow}}.$$

Substituting this into the previous display yields

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) \le \delta_i^{\leftarrow} + D_{\mathrm{KL}}(q_i^*\|\pi) + M_i\sqrt{2\delta_i^{\rightarrow}}.$$

Finally, by the definition of Δ_i,

$$D_{\mathrm{KL}}(q_i^*\|\pi) = D_{\mathrm{KL}}(P_i\|\pi) - \Delta_i.$$

Substituting this identity proves (C.1). □

Corollary C.2. Under the assumptions of Theorem C.1, if $\delta_i^{\leftarrow} + M_i\sqrt{2\delta_i^{\rightarrow}} < \Delta_i$, then

$$D_{\mathrm{KL}}(P_{i+1}\|\pi) < D_{\mathrm{KL}}(P_i\|\pi).$$

The bound decomposes the effect of projection into the ideal gain Δ_i from the exact trust-region update and the approximation cost $\delta_i^{\leftarrow} + M_i\sqrt{2\delta_i^{\rightarrow}}$. Thus, the practical update preserves the improvement whenever the projection error is smaller than the ideal gain.

C.6 Proof of Theorem 4.5

Proof. From (4.2), we have

$$\frac{dq_{i,\tau}}{dP_i}(X) = \exp\Big(\tau \log R_i(X) - \Lambda_i(\tau)\Big).$$

Since $R_i(X) = \frac{d\pi}{dP_i}(X)$, it follows that

$$\frac{d\pi}{dq_{i,\tau}}(X) = \frac{d\pi/dP_i}{dq_{i,\tau}/dP_i}(X) = \exp\big(\Lambda_i(\tau)\big) R_i(X)^{1-\tau}.$$

Therefore,

$$1 + \chi^2(\pi \| q_{i,\tau}) = \mathbb{E}_{X\sim q_{i,\tau}}\left[\left(\frac{d\pi}{dq_{i,\tau}}(X)\right)^2\right] = \exp\big(2\Lambda_i(\tau)\big)\mathbb{E}_{X\sim q_{i,\tau}}\Big[R_i(X)^{2(1-\tau)}\Big].$$

Rewriting the expectation under $q_{i,\tau}$ in terms of P_i gives

$$\mathbb{E}_{X\sim q_{i,\tau}}\Big[R_i(X)^{2(1-\tau)}\Big] = \frac{\mathbb{E}_{X\sim P_i}[R_i(X)^{2-\tau}]}{\mathbb{E}_{X\sim P_i}[R_i(X)^{\tau}]} = \exp\Big(\Lambda_i(2-\tau) - \Lambda_i(\tau)\Big),$$

and hence

$$1 + \chi^2(\pi \| q_{i,\tau}) = \exp\Big(\Lambda_i(\tau) + \Lambda_i(2-\tau)\Big).$$

Next, since $\gamma = Z\pi$, we have $\frac{d\gamma}{dq_{i,\tau}}(X) = Z\frac{d\pi}{dq_{i,\tau}}(X)$. Therefore,

$$\begin{aligned}\frac{\mathrm{Var}_{X\sim q_{i,\tau}}\left[\frac{d\gamma}{dq_{i,\tau}}(X)\right]}{Z^2} &= \mathrm{Var}_{X\sim q_{i,\tau}}\left[\frac{d\pi}{dq_{i,\tau}}(X)\right] \\ &= \mathbb{E}_{X\sim q_{i,\tau}}\left[\left(\frac{d\pi}{dq_{i,\tau}}(X)\right)^2\right] - \left(\mathbb{E}_{X\sim q_{i,\tau}}\left[\frac{d\pi}{dq_{i,\tau}}(X)\right]\right)^2.\end{aligned}$$

The second term equals to 1 because $\mathbb{E}_{X\sim q_{i,\tau}}\left[\frac{d\pi}{dq_{i,\tau}}(X)\right] = \int \pi(dX) = 1$. Thus,

$$\frac{\mathrm{Var}_{X\sim q_{i,\tau}}\left[\frac{d\gamma}{dq_{i,\tau}}(X)\right]}{Z^2} = \mathbb{E}_{X\sim q_{i,\tau}}\left[\left(\frac{d\pi}{dq_{i,\tau}}(X)\right)^2\right] - 1 = \chi^2(\pi \| q_{i,\tau}),$$

which proves (4.4).

It remains to prove monotonicity. Taking logarithms of the closed-form expression above yields

$$\log\Big(1 + \chi^2(\pi \| q_{i,\tau})\Big) = \Lambda_i(\tau) + \Lambda_i(2-\tau).$$

Differentiating with respect to τ, we obtain

$$\frac{d}{d\tau}\log\Big(1 + \chi^2(\pi \| q_{i,\tau})\Big) = \Lambda_i'(\tau) - \Lambda_i'(2-\tau).$$

Since $\Lambda_i(\tau) = \log \mathbb{E}_{X\sim P_i}[R_i(X)^{\tau}]$ is convex in τ, the derivative $\Lambda_i'(\tau)$ is nondecreasing. For $\tau \in [0,1]$, we have $\tau \le 2-\tau$, so $\Lambda_i'(\tau) - \Lambda_i'(2-\tau) \le 0$. Hence $\tau \mapsto \chi^2(\pi \| q_{i,\tau})$ is nonincreasing on $[0,1]$. If $P_i \neq \pi$, then R_i is not almost surely constant, so Λ_i is strictly convex and Λ_i' is strictly increasing. Therefore, for $\tau \in [0,1)$,

$$\Lambda_i'(\tau) - \Lambda_i'(2-\tau) < 0,$$

which implies that $\tau \mapsto \chi^2(\pi \| q_{i,\tau})$ is strictly decreasing on $[0,1)$. □

D Additional Text Generation Experiments

D.1 Baseline Implementation Details

All text-generation experiments use the same MDLM base generator trained on OpenWebText, the same 15 discriminative prompts from the PPLM benchmark, and 200 reverse diffusion steps. For each random seed, we generate 20 batches per prompt, giving 300 samples per seed. All reported numbers are averaged over seeds $\{42, 43, 44\}$.

We compare against three inference-time baselines. (i) **Base** directly samples from MDLM without reward steering and uses $K = 1$. (ii) **Best-of-N** draws K independent final candidates and selects

Table 3: TRI-TSMC architecture and main hyperparameters for discrete text generation.

Component	Setting
Twisting network	Token/time-conditioned MLP with token embedding, time embedding, and mean-pooled sequence context.
Architecture size	Embedding dimension 48, hidden dimension 192, and 64 time bins.
Output	Additive vocabulary logit bias; output layer zero-initialized.
Sampling budget	$K = 16$ particles, 3 trust-region iterations, 48 total sampled trajectories.
Reward scale	$\lambda = 10$.
Trust-region update	$\epsilon = 0.2$, empirical dual solve, weighted maximum-likelihood projection.
Optimization	Adam, learning rate 10^{-4}, 120 twist updates per iteration.
Other settings	Proposal temperature 1.0, twist logit scale 1.0, gradient clipping norm 1.0.

the one with the highest terminal reward. (iii) **FK-Steering** applies particle resampling during the reverse diffusion process under a Feynman–Kac formulation. In the main text, FK-Steering uses the single PPL reward r_{PPL} to construct local potentials, with $K \in \{16, 48\}$.

For FK-Steering and TRI-TSMC, intermediate rewards are evaluated every 20 diffusion steps. At each reward-evaluation stage, we sample $n = 4$ candidate x_0 reconstructions from the predicted logits and aggregate their rewards with log-mean-exp. Reward computation is truncated to the prompt plus the first 50 generated tokens. We set the reward scale to $\lambda = 10$ for all particle-based steering methods. For GPT-2-based evaluation, r_{PPL} is computed with GPT-2; for Qwen-based evaluation, it is computed with Qwen2.5-1.5B. The CoLA score is computed using a RoBERTa-based CoLA classifier.

D.2 Text Generation Implementation Details

For text generation, TRI-TSMC uses $K = 16$ particles and 3 trust-region iterations. Therefore, each run uses 48 sampled trajectories in total, matching the total sampled trajectories used by the $K = 48$ BoN and FK-Steering baselines. At each trust-region iteration, TRI-TSMC runs the current twisted SMC sampler, computes residual trajectory weights, solves the empirical trust-region dual, and updates the twisting function by weighted maximum likelihood.

The TRI-TSMC steering reward is a mixed reward,

$$r_{\text{mix}} = r_{\text{PPL}} + \beta r_{\text{CoLA}}.$$

For the GPT-2-based experiment, we use $\beta = 0.55$ with the `max` potential. For the Qwen2.5-1.5B-based experiment, we use $\beta = 0.56$ with the `diff` potential. In the reported runs, reward components are used without additional normalization. Given the intermediate reward estimate $\hat{r}_t$ at resampling stage t, the two potential variants are

$$\log g_t = \lambda(\hat{r}_t - \hat{r}_{t-1}) \quad (\texttt{diff}), \qquad \log g_t = \lambda \max(\hat{r}_t, \hat{r}_{t-1}) \quad (\texttt{max}).$$

The `diff` potential uses stepwise reward improvement, while the `max` potential gives a more conservative local signal.

The twisting function is a small token-level MLP that adds a logit bias to the MDLM reverse transition. Given the corrupted token sequence x_t and normalized diffusion time t, it outputs $b_\theta(x_t, t) \in \mathbb{R}^{L \times V}$ and defines the twisted proposal as

$$q_\theta(x_{t-1} \mid x_t) \propto q_{\text{base}}(x_{t-1} \mid x_t) \exp\big(b_\theta(x_t, t)\big),$$

applied position-wise on masked coordinates. The architecture and main hyperparameters are summarized in Table 3.

D.3 Qwen2.5-1.5B-based Evaluation

Implementation Details. All results are averaged over three seeds $\{42, 43, 44\}$ on 15 prompts $\times$ 20 batches per prompt. For FK-Steering and TRI-TSMC, we set the reward scale to $\lambda = 10$

and resample every 20 steps. TRI-TSMC uses $K = 16$ particles, 3 outer iterations, trust-region radius $\epsilon = 0.2$, twist learning rate 10^{-4}, 120 twist updates per iteration, and a twisting network with embedding and hidden dimensions 48 and 192, respectively. In GPT-2-based alignment, TRI-TSMC uses the max-potential and mixed reward weight $\beta = 0.55$; in Qwen2.5-1.5B-based alignment, TRI-TSMC uses the diff-potential and mixed reward weight $\beta = 0.56$. We choose different potential variants because the two reward models show different empirical behavior: GPT-2-based mixed rewards are locally noisier, for which the more conservative max-potential is more robust, while Qwen2.5-1.5B-based rewards provide a smoother stepwise signal, for which the diff-potential is more effective. All experiments were conducted on NVIDIA H100 GPUs, with each run using a single GPU.

Method	K	PPL(*) ↓	CoLA ↑	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑
Base	1	104.427	0.223	**0.594**	**0.919**	**0.941**
BoN	16	51.301	0.347	0.552	0.879	0.922
BoN	48	44.109	0.371	0.532	0.852	0.904
FK-Steering	16	17.153	0.510	0.477	0.793	0.860
FK-Steering	48	10.231	0.564	0.442	0.721	0.785
TRI-TSMC	16	**9.895**	**0.712**	0.389	0.627	0.691

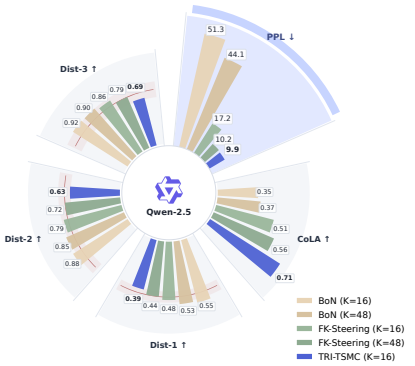


Figure 4: Alignment results on MDLM under Qwen2.5-1.5B-based evaluation. **LHS:** Quantitative results for Base, BoN, FK-Steering, and TRI-TSMC; PPL is the primary alignment objective, while CoLA and Dist-n are secondary quality indicators. **RHS:** Polar bar visualization of the same results, with the PPL sector visually emphasized. The shaded area in the secondary-metric sectors (Dist-1/2/3) indicates a $\pm 15\%$ reference band around the FK-Steering ($K = 48$) value..

Results Analysis. Figure 4 reports Qwen2.5-1.5B-based PPL reward alignment results on MDLM. TRI-TSMC again achieves the lowest perplexity, reducing PPL from 10.231 under FK-Steering ($K = 48$) to 9.895 with $K = 16$ particles. Although the PPL gain over FK-Steering is small in this setting, TRI-TSMC gives a clear improvement in CoLA, increasing it from 0.564 to 0.712. The diversity metrics are lower than FK-Steering, with Dist-1/2/3 decreasing from $0.442/0.721/0.785$ to $0.389/0.627/0.691$. This indicates that TRI-TSMC concentrates more strongly on the primary PPL objective and improves grammatical acceptability, but at the cost of reduced output diversity. Across both GPT-2-based and Qwen2.5-based evaluations, TRI-TSMC is most effective at optimizing the primary alignment objective under a smaller particle budget, while the main remaining trade-off is diversity preservation.

An interesting observation is that replacing the GPT-2 perplexity reward with the Qwen2.5 perplexity reward consistently lowers Dist-n. We interpret this as stronger reward-induced mode concentration rather than an evaluation artifact. The Qwen2.5-based mixed objective assigns higher preference to a narrower set of fluent and syntactically acceptable continuations, which reduces within-prompt lexical diversity across candidate samples. Iteration-level diagnostics (refer to Appendix D.9) support this interpretation: distinctness drops substantially from initialization to later TR iterations.

D.4 Ablation Study: FK-Steering with Mixed Rewards

The main text reports FK-Steering with the single PPL reward, following the original text-generation setup of FK-Steering (Singhal et al., 2025). In this appendix, we additionally evaluate a mixed-reward variant of FK-Steering using the same reward form as TRI-TSMC:

$$r_{\text{mix}} = r_{\text{PPL}} + \beta r_{\text{CoLA}}.$$

This ablation study tests whether the gains of TRI-TSMC can be explained simply by using a mixed reward inside FK-Steering. The mixed-reward FK variant uses the same particle resampling proce-

dure as FK-Steering; only the reward used for particle weighting and final selection is changed from r_{PPL} to r_{mix}.

Table 4: Ablation study of FK-Steering mixed-reward on MDLM. All results are averaged over 3 seeds. PPL is the primary alignment metric; CoLA and Dist-n are secondary quality indicators.

Reward model	Method	K	PPL ↓	CoLA ↑	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑
GPT-2	FK-Steering (single)	48	36.702	0.393	0.520	0.831	0.888
GPT-2	FK-Steering (mixed, $\beta = 0.40$)	48	31.134	0.710	0.525	0.820	0.876
Qwen2.5-1.5B	FK-Steering (single)	48	10.231	0.564	0.442	0.721	0.785
Qwen2.5-1.5B	FK-Steering (mixed, $\beta = 0.30$)	48	11.654	0.739	0.463	0.736	0.794

Table 4 shows that adding CoLA to FK-Steering substantially strengthens the FK baseline, especially under GPT-2-based evaluation. On GPT-2, FK-Steering with mixed reward improves both PPL and CoLA over the single-reward FK baseline. On Qwen2.5-1.5B, the mixed reward moves FK-Steering toward higher CoLA and diversity, but at the cost of worse PPL. Overall, the mixed-reward FK ablation mainly exposes a PPL–acceptability/diversity trade-off.

D.5 Single-Reward Control

The main-text comparison follows the original FK-Steering setup, which uses only the PPL reward. We therefore add an experiment in which TRI-TSMC also uses only the PPL reward. Both methods use 48 sampled trajectories per output, as shown in Table 5.

Table 5: PPL-only alignment on MDLM under GPT-2-based evaluation. Both methods use 48 total sampled trajectories per output; K denotes the number of particles per sampling round.

Reward model	Method	K	PPL ↓	CoLA ↑	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑
GPT-2	FK-Steering	48	36.702	0.393	0.520	0.831	0.888
GPT-2	TRI-TSMC (3 iterations)	16	**31.545**	0.400	0.505	0.826	0.892

Under the identical PPL-only objective and the same total trajectory budget, Table 5 shows that TRI-TSMC reduces PPL from 36.702 to 31.545, corresponding to a 14.1% improvement over FK-Steering. Since neither method optimizes CoLA in this comparison, their CoLA scores are similar, while the Dist-n scores remain broadly comparable. This result isolates the effect of iterative proposal adaptation and shows that TRI-TSMC improves the primary PPL objective even without the additional CoLA reward.

D.6 Sampling Law and Trust-Region Ablation

To directly evaluate the path-space update under its stated sampling law, we additionally implement a direct-P_i version of TRI-TSMC. During each twist-learning sweep, trajectories follow the current twisted transitions without residual-weight ancestor resampling. The residual trajectory weights are evaluated only after the trajectories are generated and are then tempered as $w_i^{\tau_i}$ for the twist update. This version directly approximates $q_i^\star(dX) \propto P_i(dX)w_i(X)^{\tau_i}$.

To isolate the contribution of the trust region, we compare two versions of direct-P_i TRI-TSMC in Table 6. Both use the same trajectories, twisting model, optimization procedure, trajectory budget, and final sampler. The no-trust-region version sets $\tau_i = 1$ and uses the raw residual weights, while the trust-region version uses the tempered weights $w_i^{\tau_i}$ obtained from the KL-constrained update. Both use the PPL-only reward.

As shown in Table 6, without the trust region, the effective sample size is approximately one trajectory out of 16, and one trajectory receives 97.3% of the regression weight. With trust-region tempering, approximately 11 trajectories contribute to the update, CV^2 decreases from 14.378 to 0.446, and the maximum weight decreases to 0.171. PPL and CoLA also improve, while diversity remains comparable.

Table 6: Effect of trust-region tempering in direct-P_i TRI-TSMC with $K = 16$. ESS/K, CV^2, and the maximum normalized weight describe the regression weights used for twist fitting.

Setting	PPL ↓	CoLA ↑	Dist-1/2/3 ↑	ESS/K ↑	CV^2 ↓	Max. weight ↓
No trust region ($\tau_i = 1$)	33.390	0.363	0.514/0.836/0.891	0.0666	14.378	0.973
Trust region (adaptive τ_i)	**31.694**	**0.397**	0.515/0.834/0.893	**0.692**	**0.446**	**0.171**

D.7 Sensitivity to Trust-Region Size, Iterations, and Particle Count

We vary one setting at a time in the GPT-2-based mixed-reward experiment. Unless varied, the configuration uses $\epsilon = 0.20$, three outer iterations, and $K = 16$ particles per iteration. The results are reported in Table 7.

Table 7: Sensitivity of TRI-TSMC on MDLM under GPT-2-based evaluation. PPL is the primary alignment metric; CoLA and Dist-n are secondary quality indicators. Changing the iteration count or K also changes the total number of sampled trajectories.

Factor	Value	PPL ↓	CoLA ↑	Dist-1 ↑	Dist-2 ↑	Dist-3 ↑
ϵ	0.10	30.582	0.718	0.493	0.779	0.833
	0.20	29.781	0.727	0.488	0.774	0.833
	0.50	31.067	0.768	0.509	0.803	0.859
Iterations	2	34.344	0.747	0.522	0.826	0.882
	3	29.781	0.727	0.488	0.774	0.833
	4	26.994	0.682	0.478	0.753	0.809
K	8	34.058	0.733	0.522	0.843	0.903
	16	29.781	0.727	0.488	0.774	0.833
	24	27.572	0.813	0.517	0.818	0.877

In Table 7, the primary PPL objective is best at the intermediate value $\epsilon = 0.20$, while making the update either more conservative or more aggressive degrades PPL. Thus, $\epsilon = 0.20$ provides the best balance between update progress and projection stability.

Increasing the iteration count consistently improves PPL because later proposals are shifted toward higher-reward regions. We use three iterations as a practical balance: it substantially improves the objective over two iterations while preserving more diversity and requiring less computation than four. PPL also improves consistently as K increases, from 34.058 at $K = 8$ to 27.572 at $K = 24$. The secondary metrics are not strictly monotone, but $K = 24$ improves CoLA and all Dist-n metrics over $K = 16$.

D.8 Runtime and Approximately Matched Compute

We profile the text-generation methods on a single NVIDIA H100 using 15 prompts and 20 batches per prompt, corresponding to 300 outputs. The time per output in Table 8 is the elapsed time of a representative single-seed job divided by 300.

Table 8: Runtime at the same 48-trajectory budget.

Method	Trajectories/output	Time/output
Best-of-N ($K = 48$)	48	10.55 s
FK-Steering ($K = 48$)	48	12.27 s
TRI-TSMC (3×16)	48	22.20 s

As Table 8 shows, TRI-TSMC takes longer at the same trajectory count because it also evaluates and optimizes the twisting function. To account for this cost, we additionally compare the methods under approximately the same end-to-end runtime and peak GPU memory in Table 9.

Table 9: PPL-only text alignment under approximately matched runtime and peak GPU memory. The direct-P_i variant samples proposal trajectories without ancestor resampling during twist learning.

Method	Traj.	Time/output	Peak memory	PPL ↓	CoLA ↑	Dist-1/2/3 ↑
Best-of-N	150	22.03 s	9.28 GiB	40.297	0.377	0.521/0.834/0.886
FK-Steering	108	20.59 s	9.28 GiB	**31.424**	0.360	0.496/0.792/0.856
TRI-TSMC (direct-P_i)	48	21.22 s	9.17 GiB	31.694	**0.397**	**0.515/0.834/0.893**

Table 10: Representative generated text samples under GPT-2-based evaluation. We compare FK-Steering ($K = 48$) with TRI-TSMC ($K = 16$) on the same prompts.

Prompt	Method	Generated text
The book	FK-Steering ($K = 48$)	*The book* will be released at 8 p.m. ET on Friday, and it will be published in Britain on Monday with Trump's blessing. Collins, however, does not aim to sell this book, but his book will . . .
	TRI-TSMC ($K = 16$)	*The book* collection, designed by Collins, will travel to museums and libraries across England. Mr Cudell will now also survive in many libraries throughout the Clifton of London. He became part of an exhibition which opened . . .
The city	FK-Steering ($K = 48$)	*The city* is a shoreline on the eastern coast of Virginia, just west of the Mississippi River, about a mile to the north and half east of Leland. The name comes from the mountains . . .
	TRI-TSMC ($K = 16$)	*The city* of Aleppo is under strict curfew and more than 4,000 people have been forced from the roads by the government's administration. Hundreds of people are also afraid to travel on the roads because of tight transport . . .
The president of the country	FK-Steering ($K = 48$)	*The president of the country* who announced the decision within an hour, has been the subject of criticism since the beginning of his administration. The judge reviewing the decision, Gerard Blanc, said that the case would . . .
	TRI-TSMC ($K = 16$)	*The president of the country*, its current leader, Mahmoud Abbas, is a frequent critic of Israel, as well as has been in favor of policies towards Iran. Despite the paradoxical appearance of "tolerance", the Palestinians insist they are . . .

Under the comparable runtime and peak-memory budgets in Table 9, FK-Steering uses 108 trajectories and Best-of-N uses 150, compared with 48 for direct-P_i TRI-TSMC. Despite using less than half as many trajectories as FK-Steering, direct-P_i TRI-TSMC obtains comparable PPL (31.694 versus 31.424), while improving CoLA from 0.360 to 0.397 and Dist-1/2/3 from 0.496/0.792/0.856 to 0.515/0.834/0.893.

D.9 Qualitative Analysis of Generated Text

We provide representative generated samples to better understand the metric trends in Table 1 and Figure 4. The examples are not intended as a human evaluation; they are selected to illustrate recurring patterns we observed in the outputs, including local fluency, malformed-token artifacts, repetition, and the quality–diversity trade-off.

Case Studies Under GPT-2 Evaluation. Table 10 compares FK-Steering and TRI-TSMC under GPT-2-based evaluation. In this setting, FK-Steering with $K = 48$ usually produces readable continuations, but the outputs often stay close to generic news-style templates or only loosely develop the prompt. For example, the continuation for "The book" is locally plausible, but quickly becomes repetitive around the word "book". The continuation for "The city" gives a geographic description, but the details are generic and weakly grounded in the prompt.

Table 11: Representative generated text samples under Qwen2.5-based evaluation. We compare FK-Steering ($K = 48$) with TRI-TSMC ($K = 16$) on the same prompts.

Prompt	Method	Generated text
The book	FK-Steering ($K = 48$)	*The book* allowing off s Acut Listnt exp league *accay offashion eker 2006.ile Sen off pThey'S 19 p Frasernt exp tearans 2006 S 2008.ileThey ...
	TRI-TSMC ($K = 16$)	*The book* has received more than 400 reviews from people across the world, has become the bestseller on Amazon, and it has been nominated for three Grammy Awards. It has read close to 650,000 times ...
The city	FK-Steering ($K = 48$)	*The city* desrent purch beam e 2011-12, 2011-12,ay 2019-20 year S 2014-15ay 2015- ...
	TRI-TSMC ($K = 16$)	*The city* of Oakland is located on the border of the city of San Francisco and the city of San Francisco Bay. It is, as of 2012, the second most populous city in America. It is also part of the ...
The president of the country	FK-Steering ($K = 48$)	*The president of the country* createdpect BaseType rout p Petut plements resist, pchoolut plements resist,ayout pchoolut p Carorecom Petut p numbers attemptedets document e p losingut pchool ...
	TRI-TSMC ($K = 16$)	*The president of the country* intends to do everything he can to avoid unprecedented condemnations from his party's allies, critics and the outside world. I will try to deal with all sides of the country. Of course, we will push ...

TRI-TSMC tends to produce more structured continuations in these examples. For "The book", the output develops a concrete setting around a collection, museums, and libraries. For "The city", the output moves toward an event-centered description involving curfew, displacement, and transportation constraints. These samples are not necessarily factually reliable, but they are more sentence-like and locally organized. This is consistent with the main GPT-2 results: TRI-TSMC improves the primary PPL objective and substantially improves CoLA, while the lower Dist-n scores indicate a more concentrated set of high-reward continuations.

Case Studies Under Qwen2.5 Evaluation. The difference is more visible under Qwen2.5-based evaluation, as shown in Table 11. FK-Steering with the single Qwen2.5 reward can select outputs with low evaluator perplexity, but some selected continuations contain malformed tokens, broken word pieces, or repeated numeric fragments. For instance, the FK-Steering output for "The book" contains fragments such as "Acut Listnt", and the output for "The city" collapses into repeated year-like strings. These artifacts suggest that optimizing the PPL reward alone can sometimes select sequences that exploit the evaluator without remaining well-formed at the surface level.

TRI-TSMC reduces this failure mode in the shown examples. The continuations for "The book", "The city", and "The president of the country" are more grammatical and easier to read, even though they are still generic and not guaranteed to be factually correct. This matches the quantitative results: TRI-TSMC improves CoLA over FK-Steering under Qwen2.5-based evaluation, but it also lowers Dist-n. In other words, the method appears to trade surface diversity for more stable, sentence-like continuations.

Overall Pattern. The qualitative examples support the same picture as the main metrics. FK-Steering with a large particle budget is effective at lowering PPL, but its selected samples can still contain generic templates, local repetition, or malformed-token artifacts, especially under the Qwen2.5 reward. TRI-TSMC shifts the selection toward continuations that are more fluent and grammatically acceptable, but it also concentrates probability mass on fewer modes. This explains why TRI-TSMC improves the primary alignment objective and CoLA, while Dist-1/2/3 decrease. Diversity preservation remains the main limitation of the current method.

Table 12: TRI-TSMC architecture and main hyperparameters for text-to-image diffusion experiments.

Component	Setting
Twisting network	Latent-space CNN conditioned on normalized diffusion time.
Architecture size	Latent channels 4, hidden channels 64, and 64 time bins.
Output	Latent noise correction $\delta_\theta(z_t, t)$; final convolution zero-initialized.
Sampler	Stochastic DDIM with 100 steps, $\eta = 1$, CFG scale 7.5.
Sampling budget	$K = 4$ particles, one trust-region update, 8 total sampled trajectories.
Resampling / reward	Resampling every 20 steps; ImageReward evaluated on decoded $\hat{x}_0$.
Potential	`diff` potential with $\lambda = 20$.
SD v1.5 setting	$\epsilon = 0.05$, twist learning rate 3×10^{-4}, 200 twist updates.
SDXL setting	$\epsilon = 0.01$, twist learning rate 2×10^{-3}, 200 twist updates, EMA twist sampling with decay 0.995.
Optimization	Adam with gradient clipping norm 1.0.

E Additional Text-to-Image Experiments

E.1 Text-to-Image Implementation Details

For text-to-image generation, TRI-TSMC uses the Stable Diffusion latent diffusion sampler as the base proposal. We use stochastic DDIM sampling with 100 denoising steps, $\eta = 1$, and classifier-free guidance scale 7.5. The steering reward is ImageReward. At each resampling stage, we decode the denoised prediction $\hat{x}_0$, evaluate ImageReward, and convert the intermediate reward into a local Feynman–Kac weight. We use $K = 4$ particles, resampling frequency 20, and one trust-region update, so each prompt has an initialization stage and one updated TRI-TSMC stage. This gives 8 total sampled trajectories, matching the $K = 8$ BoN and FK-Steering sampling budget.

The text-to-image twisting function operates in latent space. At a DDIM step, the base transition can be written as

$$z_{t-1} = \mu_{\text{base}}(z_t, t) + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

TRI-TSMC learns a latent correction $\delta_\theta(z_t, t)$ and samples from

$$z_{t-1} = \mu_{\text{base}}(z_t, t) + \sigma_t \big(\epsilon + \delta_\theta(z_t, t)\big).$$

Thus, the twisting network shifts the Gaussian noise component of the DDIM transition. The architecture and main hyperparameters are summarized in Table 12.

E.2 Additional SDXL Experiments

Table 13 reports additional text-to-image alignment results using SDXL as the base generator. IR remains the primary alignment metric. TRI-TSMC achieves the highest IR score, improving over FK-Steering ($K = 8$) from 1.238 to 1.275 and over Best-of-N ($K = 8$) from 1.195 to 1.275. This shows that the same trust-region twisting mechanism also improves the primary reward on a stronger base generator.

On secondary metrics, TRI-TSMC obtains an HPS score of 0.300, which is close to the best HPS score 0.301 from Best-of-N ($K = 8$) and higher than FK-Steering ($K = 8$). For GenEval, TRI-TSMC obtains 0.660, which is comparable to FK-Steering ($K = 4$) but lower than the larger-budget Best-of-N and FK-Steering baselines. Thus, the SDXL results show the same pattern as the SD v1.5 results: TRI-TSMC improves the IR alignment objective, while secondary metrics are also comparable with baselines

E.3 Qualitative Analysis of Generated Image

The main SD v1.5 results are reported in Table 2. The qualitative comparisons in Figures 5 and 6 provide additional examples of the same trend. Compared with the base sampler and post-training DPO reference, TRI-TSMC tends to produce images that more directly emphasize the prompt subject while remaining visually plausible. Compared with BoN and FK-Steering, TRI-TSMC often

Table 13: Main results on text-to-image alignment with SDXL. IR is the primary alignment metric. GenEval evaluates compositional prompt fidelity, and HPS evaluates human preference.

Base	Method	K	IR(*) ↑	GenEval ↑	HPS ↑
SDXL	Base	1	0.752	0.551	0.282
SDXL	DPO	1	0.913	0.586	0.299
SDXL	Best-of-N	4	1.094	0.732	0.296
SDXL	Best-of-N	8	1.195	**0.811**	**0.301**
SDXL	FK-Steering	4	1.076	0.665	0.292
SDXL	FK-Steering	8	1.238	0.723	0.298
SDXL	**TRI-TSMC**	4	**1.275**	0.660	0.300

produces samples that are more aligned with the IR-preferred interpretation of the prompt, although the examples also show that improving IR does not necessarily dominate every secondary criterion such as compositional fidelity.

In Figure 6, the initialization particles contain plausible landscape images but only weakly express the "treehouse mansion on mountain" prompt. After one trust-region twist update, the particle set shifts toward images with clearer treehouse-like structures and stronger prompt correspondence. This supports the quantitative observation in the main text: the learned twist steers the particle distribution toward higher-IR generations while maintaining reasonable visual quality and prompt alignment.

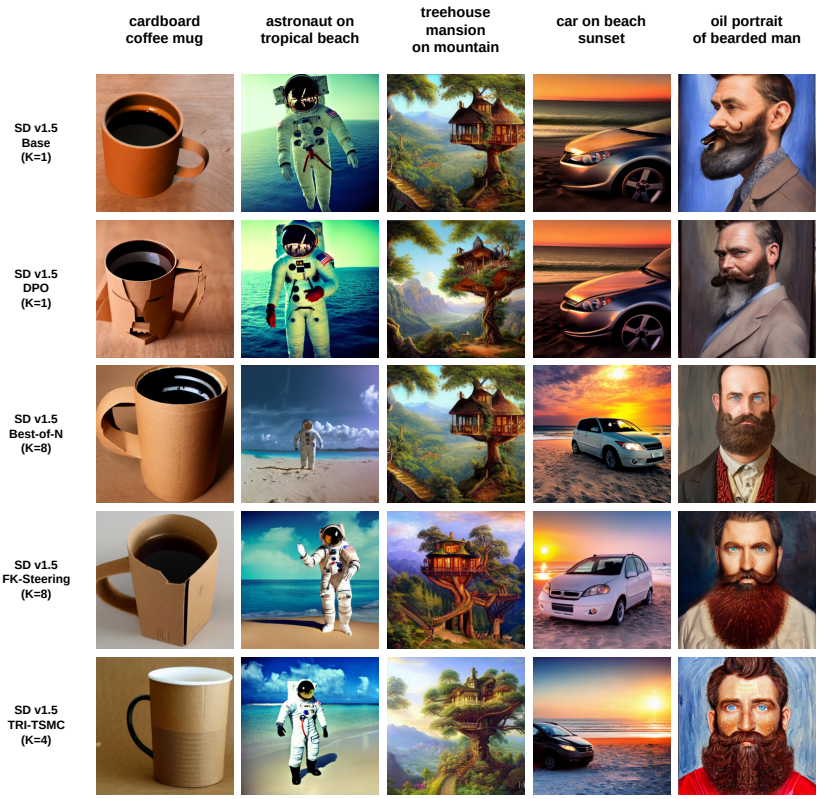


Figure 5: Qualitative comparison of text-to-image alignment methods using SD v1.5 as the base generator. The prompts are selected from the IR benchmark. Each column corresponds to a representative prompt, and each row shows samples from a different baselines including the base sampler, DPO, Best-of-N, FK-Steering, and TRI-TSMC under the indicated particle budget. TRI-TSMC produces visually competitive samples while improving prompt alignment under the same inference-time sampling budget.



Figure 6: TRI-TSMC qualitative samples for the prompt “fancy treehouse mansion on mountain.” The left four images show the initialization particles before the trust-region twist update, and the right four images show the samples after one TRI-TSMC iteration. The comparison illustrates how the learned twisting update steers the particle set toward more prompt-aligned and visually plausible generations.