BRIDGETWIST: Twisting Schrödinger Bridges for Training-Free Conditional Sampling

Mengyu Li, Qianqian Qu, Jun S. Liu*
Department of Statistics and Data Science, Tsinghua University
mengyuli@tsinghua.edu.cn quqq22@mails.tsinghua.edu.cn
junsliu@tsinghua.edu.cn

Abstract

Generative models are often required to serve many downstream conditional tasks. In the training-free setting, a pretrained unconditional model is reused as a prior and combined with a task-specific likelihood to sample from the corresponding posterior. While such training-free samplers have been developed for diffusion and flow models, their counterpart for Schrödinger bridge (SB) models remains unexplored, despite the growing appeal of SBs as flexible stochastic transport models between general source and target distributions. We propose BRIDGETWIST, a training-free conditional sampler for pretrained SB generative models, formulated as a tempered twisted sequential Monte Carlo scheme. The twist function employs a plug-in surrogate built on a closed-form endpoint predictor, which is derived by combining the velocity and score fields obtained from the SB pretraining. Since this surrogate is less reliable near the source, we temper the twist function with a time-varying exponent, downweighting its influence at early steps. We further introduce a bridge-shaped inference-time diffusion schedule that concentrates the proposal noise in the interior of the path. Telescoping importance weights then cancel all intermediate twist factors, yielding an asymptotically exact sampler for the posterior. Empirically, BRIDGETWIST outperforms training-free baselines on class-conditional and text-prompted image generation, and is competitive with specialized inverse problem solvers on inpainting, deblurring, and super-resolution.

1 Introduction

Conditional sampling draws from a generative model subject to specific constraints. In the training-free regime, a pretrained unconditional model is treated as a prior $p_\theta(x) \approx p_{\text{data}}(x)$ and combined with a tractable likelihood $L_y(x) = p(y|x)$. Bayes' rule then defines the target posterior $p_\theta(x|y) \propto p_\theta(x)L_y(x)$, allowing a single pretrained model to support many downstream conditional tasks without retraining. Different choices of L_y recover classifier-guided sampling, inverse problems, text-guided generation, and protein motif scaffolding [1–5].

For score-based diffusion and flow matching priors, training-free conditional sampling has been developed along two main lines. Gradient guidance and projection methods modify the unconditional sampling dynamics with likelihood-driven corrections at each step [2, 6–10]; they are simple and effective, but generally introduce irreducible approximation error. *Sequential Monte Carlo* (SMC) samplers instead propagate a set of particles with reweighting and resampling, yielding consistent posterior samples and often stronger performance in various conditional tasks [11, 1, 12, 3, 13, 14, 5]. Across these training-free methods, an important component is a model-specific endpoint predictor from the current intermediate state; e.g., it is used to form lookahead twists in SMC samplers.

*To whom correspondence should be addressed.

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

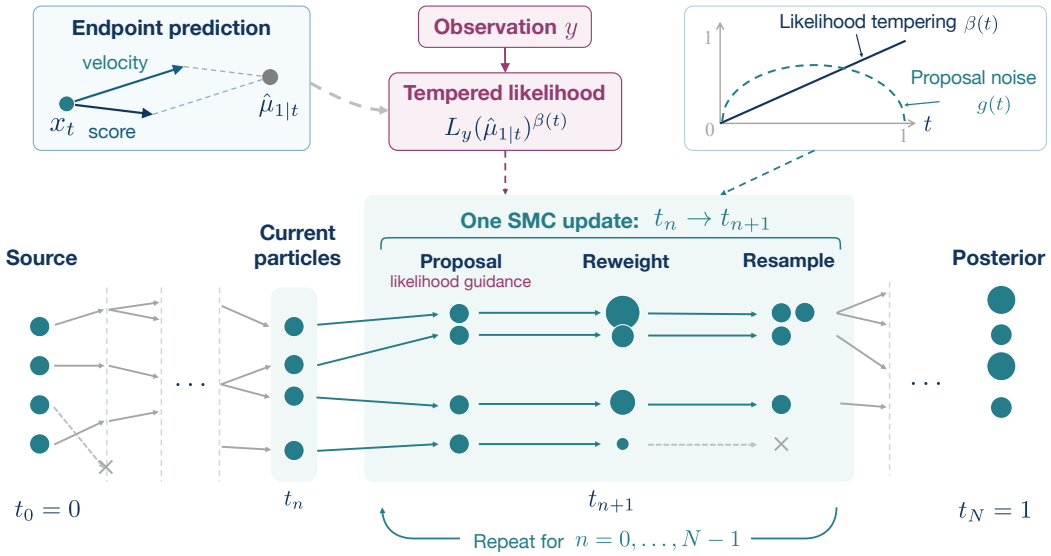


Figure 1: BRIDGETWIST overview. Schrödinger bridge's velocity and score predict endpoints for likelihood evaluation. The schedules $\beta(t)$ and $g(t)$ control likelihood tempering and proposal noise, respectively. Vertical position denotes state and circle area denotes weight. Resampling copies retain their states and are offset horizontally for visibility. The highlighted SMC update repeats along the time grid, and the terminal weighted particles approximate the target conditional distribution.

Schrödinger bridge (SB) generative models [15–18] provide a recent alternative to diffusion and flow priors. An SB connects a source $X_0 \sim q_0$ to a target $X_1 \sim q_1$; among all path laws with these source and target marginals, it selects the one that minimizes the Kullback–Leibler (KL) divergence to a prescribed reference law. This combination of a general source and stochastic transport is useful for image generation [15, 19, 20], image restoration [21, 22], cell trajectory inference [18, 23], and other tasks. Existing SB-based conditional methods [24, 21, 25, 26] train a task-specific network for each likelihood family. Training-free conditional sampling that reuses a pretrained unconditional SB prior remains unexplored.

Under a Brownian reference with diffusion scale $\sigma > 0$, an SB conditioned on its endpoints (X_0, X_1) is a Brownian bridge. At each $t \in (0, 1)$, we can therefore write $X_t = (1-t)X_0 + tX_1 + \varepsilon_t$, where $\varepsilon_t \sim \mathcal{N}(0, \sigma^2 t(1-t)I)$ is independent of (X_0, X_1). For endpoint prediction, diffusion samplers use score-based denoising, and flow samplers use velocity-based extrapolation (Table 1). For SBs, however, score-only denoising estimates the interpolation $(1-t)X_0 + tX_1$ instead of X_1, while velocity-only extrapolation omits a score correction. Our key observation is that the probability-flow velocity $v_t^\star$ and marginal score $s_t^\star$ are complementary, and combining these two fields gives the endpoint conditional mean: $\mathbb{E}[X_1|X_t = x] = x + (1-t)v_t^\star(x) + (\sigma^2/2)(1-t)s_t^\star(x)$ (Prop. 1). Their neural approximations can be obtained through score and flow matching [18] or from learned forward and backward SB drifts [15, 16].

Table 1: Training-free conditional sampling landscape from an endpoint prediction view.

Model class	What is X_t?	Endpoint signal	Representative samplers
Diffusion	noisy observation of X_1	score (Tweedie)	DPS [2], TDS [1], FPS [3]
Flow matching	deterministic interpolation of (X_0, X_1)	velocity (Euler)	FMPS [6], TFTF [5]
Schrödinger bridge	noisy interpolation of (X_0, X_1)	**velocity + score** (Prop. 1)	**BRIDGETWIST (this work)**

Using these pretrained fields, we propose BRIDGETWIST, a tempered twisted SMC sampler (Figure 1). The ideal twist, i.e., the predictive likelihood $\mathbb{E}[L_y(X_1)|X_t]$, is unavailable analytically, and using pilot rollouts to estimate it [27, 28] requires many network evaluations per particle and step. We instead use a plug-in twist $L_y(\hat{X}_1(X_t))$ to approximate, where $\hat{X}_1(X_t)$ is the velocity–score endpoint prediction. This twist discards uncertainty in X_1 given X_t and can be over-confident near the source. We therefore temper its influence with a time-varying temperature coefficient. In an isotropic linear-Gaussian model, the tempered plug-in twist can recover the exact twist (Remark 1).

With velocity and score fields, we can vary the diffusion coefficient $g(t)$ at inference time while preserving the SB marginals (Prop. 2) [18, 29]. We use this flexibility to introduce a bridge-shaped schedule $g(t) \propto \sqrt{t(1-t)}$, following the time dependence of the Brownian bridge standard deviation. This schedule concentrates noise in the middle of the sampling path. It reduces noise near the source, where guidance is weak, and near the target, where additional noise can disrupt particles already aligned with the conditioning information.

Our contributions. We believe that the following progresses are novel.

- **Training-free SB conditioning.** BRIDGETWIST is, to our knowledge, the first training-free conditional sampler for pretrained Schrödinger bridges, supporting arbitrary differentiable likelihoods.
- **Tempered plug-in twist.** We construct a lookahead twist from the velocity–score endpoint predictor and temper its influence along the path, without changing the final target conditional distribution.
- **Noise schedule and consistency.** We introduce a bridge-shaped diffusion schedule and importance correction; Theorem 1 bounds the L_2 error for K particles by order $K^{-1/2}$ relative to the learned and discretized posterior.
- **Plug-and-play for diverse tasks.** BRIDGETWIST achieves leading performance on various image generation and restoration tasks without task-specific retraining.

2 Background

Notation. Random objects are denoted by uppercase letters (e.g., X_t), and their realizations by lowercase ones (e.g., x_t). We write $\mathcal{N}(\mu, \Sigma)$ for a Gaussian density and I for the identity matrix. The time index is $t \in [0, 1]$, and its discretization is t_n for $n \in \{0, \dots, N\}$ with $t_0 = 0$ and $t_N = 1$.

2.1 Schrödinger bridge generative models

Let $\Omega := C([0, 1], \mathbb{R}^d)$ be the space of continuous paths in $\mathbb{R}^d$, and let q_0 and q_1 be the source (or noise) and target (or data) distributions. Let Q be a Brownian reference path measure under which $\mathrm{d}X_t = \sigma \, \mathrm{d}W_t$, where $\sigma > 0$ and W_t is the standard Brownian motion. The *Schrödinger bridge* problem [30, 31] seeks the path measure on Ω closest to Q in Kullback–Leibler divergence subject to the prescribed source and target marginals,

$$P^\star = \underset{P:\, p_0 = q_0,\, p_1 = q_1}{\arg\min} \mathrm{KL}(P \| Q),$$

where p_t denotes the marginal of P at time t. The bridge $P^\star$ admits a forward stochastic differential equation (SDE),

$$\mathrm{d}X_t = b_t^\star(X_t) \, \mathrm{d}t + \sigma \, \mathrm{d}W_t, \qquad X_0 \sim q_0. \tag{1}$$

The associated probability-flow ordinary differential equation (ODE) [32, 18] has the same marginal distributions p_t as Eq. (1):

$$\mathrm{d}X_t = v_t^\star(X_t) \, \mathrm{d}t, \qquad X_0 \sim q_0. \tag{2}$$

The drift $b^\star$, velocity $v^\star$, and score $s^\star$ are linked by

$$b_t^\star(x) = v_t^\star(x) + \tfrac{1}{2}\sigma^2 s_t^\star(x), \qquad s_t^\star(x) = \nabla \log p_t(x).$$

Learning the Schrödinger bridge. Earlier iterative schemes [15, 16] learn the bridge through repeated forward and backward SDE simulations and updates to drift networks. This repeated SDE simulation makes them computationally expensive. Simulation-free score and flow matching [18] circumvents the need to simulate the learned SDE during training. Instead, it draws endpoint pairs (X_0, X_1) from an entropic optimal transport (OT) coupling and, for each pair, samples an intermediate state X_t from the corresponding Brownian bridge. Then the conditional bridge velocity and score at (t, X_t) serve as regression targets for (v_θ, s_θ), neural approximations of $(v^\star, s^\star)$. Appendix B details the pretraining parameterization used in our experiments.

Given (v_θ, s_θ) and a diffusion schedule $g : [0, 1] \to \mathbb{R}_{\geq 0}$, to sample from q_1, one draws $X_0 \sim q_0$ and simulates

$$\mathrm{d}X_t = b_{\theta,g}(t, X_t) \, \mathrm{d}t + g(t) \, \mathrm{d}W_t, \qquad b_{\theta,g}(t, x) = v_\theta(t, x) + \tfrac{1}{2} g(t)^2 s_\theta(t, x). \tag{3}$$

The choice of g is an inference-time degree of freedom; under exact fields, any continuous $g \geq 0$ preserves the SB marginals p_t [18, 29]. We write p_θ for the endpoint marginal of X_1 under learned fields.

2.2 Sequential Monte Carlo and twisted SMC

Sequential Monte Carlo is a Bayesian computational framework that approximates a sequence of path-space distributions $\{\pi_n\}_{n=0}^N$ using weighted particles [33–35]. On a grid $0 = t_0 < \cdots < t_N = 1$, we write $x_{t_{0:n}} := (x_{t_0}, \ldots, x_{t_n})$ for the discretized trajectory up to t_n. The framework specifies a sequence of unnormalized targets $\{\gamma_n\}_{n=0}^N$, with normalized densities $\pi_n(x_{t_{0:n}}) \propto \gamma_n(x_{t_{0:n}})$, and proposal kernels $\{q_n\}_{n=0}^N$. At step $n = 0$, K particles $\{X_{t_0}^{(k)}\}_{k=1}^K$ are drawn from q_0 and assigned normalized weights $W_0^{(k)} \propto \gamma_0(X_{t_0}^{(k)})/q_0(X_{t_0}^{(k)})$. For each subsequent step $n = 1, \ldots, N$, the algorithm

- *resamples* the previous-step particles in proportion to their weights;
- *propagates* each resampled particle through $q_n(\cdot \mid x_{t_{n-1}})$; and
- *reweights* by the incremental weight ratio $\omega_n(x_{t_{0:n}}) = \gamma_n(x_{t_{0:n}})/[\gamma_{n-1}(x_{t_{0:n-1}})q_n(x_{t_n}|x_{t_{n-1}})]$.

We write $\pi_n(\mathrm{d}x_{t_n})$ for the marginal of π_n at time t_n. Under standard regularity, the weighted empirical measure at t_N converges weakly to $\pi_N(\mathrm{d}x_{t_N})$ as $K \to \infty$ [36, 35].

For conditional sampling, the path target is the prior tilted by the terminal likelihood, $\Pi^\star(\mathrm{d}x_{0:1}) \propto P_{\theta,g}(\mathrm{d}x_{0:1})\, L_y(x_1)$, where $P_{\theta,g}$ is the law of Eq. (3). A naive importance sampler that draws full trajectories from $P_{\theta,g}$ and reweights by $L_y(X_1)$ can suffer severe weight degeneracy when the conditioning is sharp. *Twisted SMC* [37, 35] mitigates this by inserting intermediate twisting functions, also called lookahead functions. Under the learned prior $P_{\theta,g}$, the ideal twist is the predictive likelihood

$$h_t(x) := \mathbb{E}[L_y(X_1) \mid X_t = x], \qquad t \in [0, 1]. \tag{4}$$

This ideal twist is intractable because it integrates over all future trajectories, and direct Monte Carlo estimation is prohibitively expensive. Existing twisted SMC samplers therefore approximate h_t with model-specific surrogates, using Tweedie's endpoint mean inside the likelihood for diffusion priors [1], and the Euler one-step endpoint for flow priors [5]. Section 3 develops the counterpart for Schrödinger bridges.

3 Method

We consider conditioning information y supplied through a differentiable likelihood $L_y(x) = p(y|x)$, possibly known up to a normalizing constant. Conditional sampling aims to draw from the endpoint posterior $\pi^\star(x_1) \propto p_\theta(x_1)L_y(x_1)$. This endpoint posterior is the terminal marginal of the path posterior $\Pi^\star$ defined in Section 2.2.

Section 3.1 constructs a tractable surrogate for the ideal twist $h_t(x) = \mathbb{E}[L_y(X_1)|X_t = x]$. Section 3.2 develops the BRIDGETWIST sampler by combining temporal tempering, guided proposals, a bridge-shaped diffusion schedule, and importance correction. Section 3.3 bounds the resulting finite-particle approximation error.

3.1 Plug-in twist from the SB endpoint predictor

We first construct a first-moment surrogate for the ideal twist h_t in (4) from the SB velocity and score and use it as the lookahead of a tempered twisted SMC sampler.

With a Brownian reference, the SB in Eq. (1) conditioned on (X_0, X_1) is a Brownian bridge, that is, $X_t|(X_0, X_1) \sim \mathcal{N}(M_t, \sigma^2 t(1-t)I)$, where $M_t = (1-t)X_0 + tX_1$ is the linear interpolation between the endpoints. Writing $D = X_1 - X_0$ for the endpoint displacement gives $X_1 = M_t + (1-t)D$. The conditional means of M_t and D given X_t admit closed-form expressions in terms of the velocity and score fields and combine to yield the endpoint conditional mean.

Proposition 1. *Suppose the SB marginal density p_t is continuously differentiable on $\mathbb{R}^d$ for every $t \in (0, 1)$. Under a Brownian-reference SB with velocity $v^\star$ and score $s^\star$ as in Eq. (2), for every $t \in (0, 1)$ and $x \in \mathbb{R}^d$,*

$$\mathbb{E}[M_t \mid X_t = x] = x + \sigma^2 t(1-t)s_t^\star(x), \tag{5}$$

$$\mathbb{E}[D \mid X_t = x] = v_t^\star(x) + \tfrac{1-2t}{2}\sigma^2 s_t^\star(x), \tag{6}$$

$$\mu_{1|t}^\star(x) := \mathbb{E}[X_1 \mid X_t = x] = x + (1-t)v_t^\star(x) + \tfrac{\sigma^2}{2}(1-t)s_t^\star(x). \tag{7}$$

Proposition 1 serves two purposes: (i) Eq. (7) provides a closed-form endpoint predictor, that is, a linear extrapolation $\mu^\star_{1|t}(x) = x + (1-t)b^\star_t(x)$ along the SB forward drift $b^\star$. (ii) More importantly, identities (5)–(6) explain why neither single channel suffices on its own: a score-only Tweedie predictor in (5) recovers the interpolation instead of the endpoint, while a velocity-only Euler predictor $x + (1-t)v^\star_t(x)$ omits the score correction in (7); Section 4.3 ablates both. The stochastic interpolant framework of Albergo et al. [29] provides velocity and score relations that leave the endpoint mean implicit. Proposition 1 explicitly recovers this mean for the Brownian bridge interpolant. The proof is in Appendix A.1.

With pretrained neural fields v_θ and s_θ, we obtain a practical endpoint predictor

$$\hat{\mu}_{1|t}(x) := x + (1-t)v_\theta(t,x) + \tfrac{\sigma^2}{2}(1-t)s_\theta(t,x), \tag{8}$$

and use it to define the twist surrogate

$$\hat{h}_t(x) := L_y(\hat{\mu}_{1|t}(x)) \quad \text{for} \quad t \in (0,1), \qquad \hat{h}_1(x) = L_y(x).$$

3.2 The BRIDGETWIST sampler

Fix a grid $0 = t_0 < t_1 < \cdots < t_N = 1$ with $\Delta_n = t_n - t_{n-1}$, and use the Euler–Maruyama prior kernel

$$M_n(x_{t_{n-1}}, \mathrm{d}x_{t_n}) := \mathcal{N}\big(x_{t_n};\, x_{t_{n-1}} + b_{\theta,g}(t_{n-1}, x_{t_{n-1}})\Delta_n,\, g(t_{n-1})^2\Delta_n\, I\big)\mathrm{d}x_{t_n}, \tag{9}$$

with $b_{\theta,g}$ and g defined in Eq. (3). The corresponding path law is $P^{(n)}_{\theta,g}(\mathrm{d}x_{t_{0:n}}) := q_0(\mathrm{d}x_{t_0})\prod_{i=1}^n M_i(x_{t_{i-1}}, \mathrm{d}x_{t_i})$ for $n \le N$.

Tempered intermediate targets. Even with an exact conditional mean, the plug-in twist discards the conditional spread of $X_1|X_t$, so in general

$$L_y\big(\mathbb{E}[X_1 \mid X_t = x]\big) \neq \mathbb{E}[L_y(X_1) \mid X_t = x] = h_t(x).$$

Since this spread is large at early times, evaluating the likelihood at a single endpoint prediction can make the twist over-confident (Remark 1). Therefore, we temper the plug-in twist, in the spirit of annealed importance sampling [38, 39], by the exponent $\beta(t) = t^a$ with $a \ge 1$, so that $\beta(0) = 0$ and $\beta(1) = 1$. Letting $\beta_n = \beta(t_n)$ and $\hat{h}_n = \hat{h}_{t_n}$, the intermediate (unnormalized) path targets are

$$\gamma^\beta_n(\mathrm{d}x_{t_{0:n}}) := P^{(n)}_{\theta,g}(\mathrm{d}x_{t_{0:n}})\,\hat{h}_n(x_{t_n})^{\beta_n}, \qquad \pi^\beta_n = \gamma^\beta_n / Z^\beta_n.$$

At $n = N$, $\beta_N = 1$ and $\hat{h}_N = L_y$, so $\gamma^\beta_N(\mathrm{d}x_{t_{0:N}}) = P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}})L_y(x_{t_N})$. Let $p^{(N)}_\theta(\mathrm{d}x_{t_N})$ denote the endpoint marginal of the discretized prior path law $P^{(N)}_{\theta,g}$. Normalizing γ^β_N and integrating out $x_{t_{0:N-1}}$ then gives the desired endpoint posterior $\pi^\beta_N(\mathrm{d}x_{t_N}) = \pi^{\star,(N)}(\mathrm{d}x_{t_N}) \propto p^{(N)}_\theta(\mathrm{d}x_{t_N})L_y(x_{t_N})$.

Guided proposal and telescoping weights. The continuous-time Doob h-transform adds $g(t)^2\nabla_x \log h_t(x)$ to the prior drift [37, 1, 35]. Replacing the ideal twist h by the tempered plug-in twist $\hat{h}^\beta$, we obtain the guided proposal $q_n = \mathcal{N}(m_{q,n}, \Sigma_{q,n})$ with covariance $\Sigma_{q,n} = (g(t_{n-1})^2\Delta_n + \varepsilon)I$, $\varepsilon > 0$, and mean

$$m_{q,n}(x_{t_{n-1}}) = x_{t_{n-1}} + \big[b_{\theta,g}(t_{n-1}, x_{t_{n-1}}) + w_g\, g(t_{n-1})^2\beta_{n-1}\nabla_x \log \hat{h}_{n-1}(x_{t_{n-1}})\big]\Delta_n, \tag{10}$$

where $w_g \ge 0$ is a guidance strength, and the inflation εI makes the density ratio M_n/q_n uniformly bounded in x_{t_n}. The incremental weight

$$\omega_n(x_{t_{n-1}}, x_{t_n}) := \frac{M_n(x_{t_{n-1}}, x_{t_n})\,\hat{h}_n(x_{t_n})^{\beta_n}}{q_n(x_{t_n} \mid x_{t_{n-1}}, y)\,\hat{h}_{n-1}(x_{t_{n-1}})^{\beta_{n-1}}} \tag{11}$$

corrects the guided proposal toward π^β_n, and multiplying Eq. (11) over n gives the telescoping identity (Eq. (24) in Appendix A.3). All intermediate twist factors thus cancel in the telescoping product. Therefore, although tempering changes both the intermediate targets and the proposal drift, the endpoint target is unchanged.

Algorithm 1 BRIDGETWIST: tempered twisted SMC for Schrödinger bridges

Input: Pretrained fields (v_θ, s_θ); source q_0; likelihood L_y; grid $t_{0:N}$; diffusion schedule g; guidance strength w_g; tempering exponent a; number of particles K; ESS threshold $\rho \in (0, 1]$.

Output: Weighted endpoint particles $\{(X_{t_N}^{(k)}, W_N^{(k)})\}_{k=1}^K$ approximating $\pi^{\star,(N)}$.

Set $\beta_n = t_n^a$ and $b_{\theta,g}(t,x) = v_\theta(t,x) + \frac{1}{2}g(t)^2 s_\theta(t,x)$.
Sample $X_{t_0}^{(k)} \sim q_0$ and set $\log W_0^{(k)} = \beta_0 \log \hat{h}_0(X_{t_0}^{(k)})$ for $k = 1, \dots, K$; normalize $\{W_0^{(k)}\}_{k=1}^K$.
for $n = 1, \dots, N$ **do**
    If $\mathrm{ESS}_{n-1} := 1/\sum_k (W_{n-1}^{(k)})^2 < \rho K$, resample ancestors $A_{n-1}^{(1:K)}$ and reset $\log W_{n-1}^{(k)} = -\log K$; otherwise set $A_{n-1}^{(k)} = k$. ▷ resampling
    for $k = 1, \dots, K$ **do** ▷ propagation and reweighting
        Let $\bar{X}_{t_{n-1}}^{(k)} = X_{t_{n-1}}^{(A_{n-1}^{(k)})}$ and evaluate $\nabla_x \log \hat{h}_{n-1}(\bar{X}_{t_{n-1}}^{(k)})$.
        Sample $X_{t_n}^{(k)} \sim q_n(\cdot|\bar{X}_{t_{n-1}}^{(k)}, y)$ from Eq. (10).
        Update $\log W_n^{(k)} = \log W_{n-1}^{(k)} + \log \omega_n(\bar{X}_{t_{n-1}}^{(k)}, X_{t_n}^{(k)})$ by Eq. (11).
    end for
    Normalize $\{W_n^{(k)}\}_{k=1}^K$.
end for
return $\{(X_{t_N}^{(k)}, W_N^{(k)})\}_{k=1}^K$.

Remark 1 (Tempering as variance matching). Suppose $X_1|X_t = x \sim \mathcal{N}(\mu_{1|t}^\star(x), \sigma_{1|t}^2 I)$ and $L_y(x) = \mathcal{N}(y; x, \sigma_y^2 I)$. Then $h_t(x) = \mathcal{N}(y; \mu_{1|t}^\star(x), (\sigma_y^2 + \sigma_{1|t}^2) I)$. For an exact conditional mean $\hat{\mu}_{1|t} = \mu_{1|t}^\star$, variance matching gives

$$\beta^\star(t) = \frac{\sigma_y^2}{\sigma_y^2 + \sigma_{1|t}^2}, \qquad \hat{h}_t(x)^{\beta^\star(t)} \propto h_t(x).$$

With $w_g = 1$ and $\beta(t) = \beta^\star(t)$, the guidance in (10) recovers the exact h-transform drift correction. As $\sigma_{1|t}^2$ decreases toward the endpoint, $\beta^\star(t)$ increases to one, motivating $\beta(t) = t^a$ as a monotone surrogate that requires no variance estimate. Appendix A.4 gives the proof and the extension to observations of a subset of coordinates.

Bridge-shaped diffusion schedule. At inference time, g is flexible (Section 2.1). A constant $g(t) \equiv \sigma$ matches the Brownian reference but allocates a uniform proposal variance $\sigma^2 \Delta_n$ across n. This is mismatched with the conditional bridge variance $\sigma^2 t(1-t)$, which instead vanishes at both endpoints, and the excess proposal noise can increase the variance of importance weights near $t \in \{0, 1\}$. We therefore introduce a bridge-shaped schedule $g(t) = c_g \sigma \sqrt{t(1-t) + \delta_g}$, where $c_g > 0$ is an inference diffusion scale and the small $\delta_g > 0$ keeps g positive at $t \in \{0, 1\}$.

Remark 2 (Interaction between tempering and diffusion). The exponent $\beta(t)$ sets how much the sampler trusts the plug-in lookahead, while $g(t)$ allocates the proposal noise along the path. Small $g(t)$ is intended to limit unguided perturbations near $t = 0$, where the source already supplies particle diversity, and to avoid disturbing particles aligned with the likelihood near $t = 1$. Larger interior noise supports exploration. Ignoring the small floor δ_g, the two schedules combine in Eq. (10) through the scalar guidance coefficient $g(t)^2 \beta(t) \propto t^{a+1}(1-t)$, which peaks at $t = (a+1)/(a+2)$, or $t = 2/3$ for the default $a = 1$.

Algorithm and complexity. Algorithm 1 summarizes the proposed BRIDGETWIST sampler, with stratified resampling triggered when the effective sample size (ESS) falls below ρK. Each step requires one forward evaluation of (v_θ, s_θ) per particle and one backward pass through the endpoint predictor and the likelihood, so the total cost is $\mathcal{O}(KN)$ network evaluations and $\mathcal{O}(KN)$ likelihood gradients, the same order as gradient-guided samplers [2, 6] and twisted SMC samplers [1, 5].

3.3 Finite-particle accuracy

Let $h_n(x) := \mathbb{E}[L_y(X_1)|X_{t_n} = x]$ and $P_{n,N}^y(x, \mathrm{d}z) := \mathbb{P}(X_1 \in \mathrm{d}z | X_{t_n} = x, y)$ denote the exact twist and the posterior transition kernel from step n to step N under the discretized prior. Write $\mathrm{osc}(f) := \sup f - \inf f$ and $\varrho(P) := \sup_{\mathrm{osc}(f)\le 1} \mathrm{osc}(Pf) \in [0, 1]$ for the oscillation of f and the

contraction coefficient of a Markov kernel P, respectively. The norm $\|\cdot\|_{L_2}$ denotes the root mean square over the sampler's randomness.

Theorem 1. *Under the regularity conditions stated in Appendix A.3, for every bounded measurable $\varphi : \mathbb{R}^d \to \mathbb{R}$ and every $K \geq 1$, the weighted empirical measure produced by Algorithm 1 with stratified resampling satisfies*

$$\Big\| \sum_{k=1}^{K} W_N^{(k)} \varphi(X_{t_N}^{(k)}) - \mathbb{E}_{\pi^{\star,(N)}}[\varphi] \Big\|_{L_2} \leq \frac{C_N}{\sqrt{K}} \operatorname{osc}(\varphi), \quad C_N = 2 \sum_{n=0}^{N} \alpha_n \, R_{n,N}^2 \, \varrho(P_{n,N}^y), \tag{12}$$

where $\alpha_n < \infty$ is independent of K and controls the one-step sampling error, and $R_{n,N} := \sup_x\{h_n(x)/\hat{h}_n(x)^{\beta_n}\}/\inf_x\{h_n(x)/\hat{h}_n(x)^{\beta_n}\}$ measures the relative twist mismatch.

The endpoint predictor and tempering directly influence $R_{n,N}$, which equals one when $\hat{h}_n^{\beta_n} \propto h_n$; Remark 1 gives the corresponding Gaussian example. The local error constant α_n reflects the guided proposal, diffusion schedule, and resampling threshold ρ, while $\varrho(P_{n,N}^y)$ is the contraction coefficient of $P_{n,N}^y$. For a fixed grid, Eq. (12) gives L_2 convergence of the weighted estimates to expectations under $\pi^{\star,(N)}$, the posterior induced by the learned and discretized prior, at the rate of $K^{-1/2}$.

4 Experiments

We compare BRIDGETWIST with flow matching posterior sampling (FMPS) [6], training-free targeted flow (TFTF) [5], twisted diffusion sampler (TDS) [1], diffusion posterior sampling [2] adapted to the SB prior (DPS-SB), and terminal importance sampling (IS). MNIST provides a sanity check [40]; CIFAR-10 and 256×256 CelebA-HQ provide the main image benchmarks [41, 42]. Pretraining and implementation details are in Appendices B–C; additional results are in Appendix D. The code is publicly available at `https://github.com/Mengyu8042/BridgeTwist`.

4.1 Toy example

We first evaluate BRIDGETWIST on a two-dimensional toy example [5]. The target distribution is the Gaussian mixture $q_1 = 0.75\mathcal{N}((-\sqrt{2}, -\sqrt{2}), I/4) + 0.25\mathcal{N}((\sqrt{2}, \sqrt{2}), I/4)$, the source is $q_0 = \mathcal{N}(0, I)$, and the Brownian reference has $\sigma = 1$. We condition on $y = (0, 0)$ through the Gaussian likelihood $L_y(x_1) = \mathcal{N}(y; x_1, I/4)$. In this setting, the target posterior is available analytically as $\pi^\star(x_1) = 0.75\mathcal{N}(x_1; (-\sqrt{2}/2, -\sqrt{2}/2), I/8) + 0.25\mathcal{N}(x_1; (\sqrt{2}/2, \sqrt{2}/2), I/8)$. All velocity and score fields are computed in closed form, so this experiment isolates the conditional sampler from neural network approximation errors (Appendix C.1).

We use $N = 100$ discretization steps and $K \in \{100, 200, 500, 1000, 2000\}$ particles, and report the maximum mean discrepancy (MMD) to the analytic posterior over 50 repetitions. The left panel of Figure 2 shows that BRIDGETWIST achieves the lowest MMD. Its empirical MMD^2 decays approximately as K^{-1}, with a fitted log-log slope of -1.01. SMC-based methods improve as K increases, with BRIDGETWIST outperforming both flow-based TFTF and diffusion-based TDS. FMPS and DPS-SB plateau at higher errors because they use approximate guidance without importance weight correction, leaving bias that persists as K increases. The right panel indicates that BRIDGETWIST closely matches the two posterior modes and their relative mass.

4.2 Image conditional sampling

For a fair comparison, all methods use uniform grids with $N = 800$ steps. BRIDGETWIST, DPS-SB, and IS share the Schrödinger bridge prior; FMPS and TFTF share the flow prior; and TDS uses its native DDPM (MNIST) and iDDPM (CIFAR-10) priors.

Class-conditional sampling. A frozen classifier defines $L_y(x_1) = p_\psi(y|x_1)$ on MNIST and CIFAR-10. We measure accuracy with the likelihood classifier (ACC_L) and an independent external classifier (ACC_E). Kernel inception distance (KID) [43], Density, and Coverage [44] evaluate sample quality and coverage against real images within each class. Table 2 shows that BRIDGETWIST combines high external accuracy with low KID and broad coverage; Figure 3 illustrates class alignment and diversity.

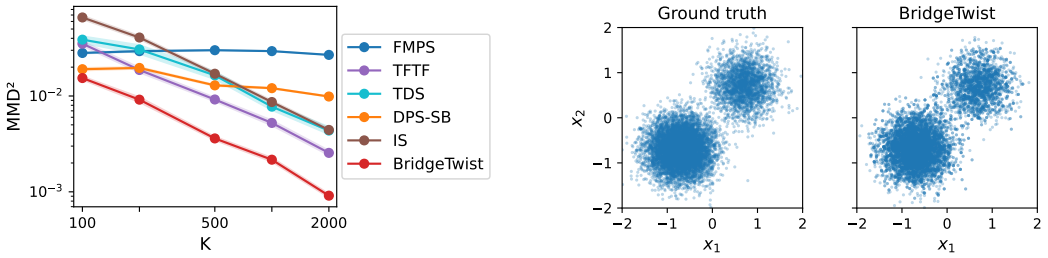


Figure 2: Toy benchmark. Left: MMD^2 to the analytic posterior as the number of particles K increases. Right: ground-truth posterior samples and BRIDGETWIST samples, with 10^4 samples in each panel.

Table 2: Class-conditional metrics at $K = 64$. Dens./Cov. denote Density/Coverage. Results are averaged over 100 random seeds. Bold marks the best value in each column.

	MNIST					CIFAR-10				
Method	$\text{ACC}_L \uparrow$	$\text{ACC}_E \uparrow$	KID $\downarrow$	Dens. $\uparrow$	Cov. $\uparrow$	$\text{ACC}_L \uparrow$	$\text{ACC}_E \uparrow$	KID $\downarrow$	Dens. $\uparrow$	Cov. $\uparrow$
FMPS	0.997	0.453	0.0202	0.026	0.007	0.899	0.514	0.1807	0.504	0.390
TFTF	0.977	0.910	0.0018	0.954	0.534	0.986	0.934	0.0572	1.161	**0.600**
TDS	0.992	0.942	0.0015	0.923	0.415	0.947	0.869	0.129	1.159	0.342
DPS-SB	**0.999**	0.619	0.0010	0.633	0.495	0.978	0.585	0.1051	0.410	0.302
IS	0.997	0.920	0.0007	1.001	0.146	0.971	0.865	0.0744	0.755	0.113
BRIDGETWIST	**0.999**	**0.952**	**0.0003**	**1.154**	**0.617**	**0.997**	**0.979**	**0.0403**	**1.288**	0.596



Figure 3: CIFAR-10 airplane samples at $K = 16$, after resampling based on the terminal weights.

Table 3 compares methods at equal total numbers of backbone function evaluations (NFEs), counting each forward or backward pass once per particle. Baselines adjust K to fit the NFE budgets $\{B_1, B_2, B_3, B_4\}$, defined by BRIDGETWIST at $K \in \{4, 16, 64, 256\}$. FMPS leads at the smallest budget, while BRIDGETWIST performs the best at all the larger budgets. Appendix D.1 compares runtime and memory usage.

Table 3: CIFAR-10 ACC_E at matched total NFEs. Subscripts give standard deviations across repetitions. Bold marks the best value.

Method \ total NFE	$B_1 = 6{,}408$	$B_2 = 25{,}632$	$B_3 = 102{,}528$	$B_4 = 410{,}112$
FMPS	$\mathbf{0.775}_{\pm 0.016}$	$0.813_{\pm 0.015}$	$0.814_{\pm 0.009}$	$0.814_{\pm 0.004}$
TFTF	$0.555_{\pm 0.145}$	$0.770_{\pm 0.116}$	$0.936_{\pm 0.054}$	$0.961_{\pm 0.032}$
DPS-SB	$0.613_{\pm 0.064}$	$0.606_{\pm 0.034}$	$0.601_{\pm 0.025}$	$0.586_{\pm 0.008}$
IS	$0.568_{\pm 0.058}$	$0.815_{\pm 0.059}$	$0.866_{\pm 0.021}$	$0.885_{\pm 0.012}$
TDS	$0.585_{\pm 0.120}$	$0.638_{\pm 0.098}$	$0.875_{\pm 0.113}$	$0.918_{\pm 0.059}$
BRIDGETWIST	$0.630_{\pm 0.112}$	$\mathbf{0.914}_{\pm 0.061}$	$\mathbf{0.979}_{\pm 0.009}$	$\mathbf{0.984}_{\pm 0.006}$

Text-prompted image generation. On CelebA-HQ, text prompts specify facial attributes, and a CLIP energy defines the text likelihood $L_y(x_1) \propto \exp\{\lambda\ \text{sim}(\phi_I(x_1), \phi_T(y))\}$ [45]. We evaluate the CLIP similarity under the likelihood model (CLIP_L), all-attributes accuracy from an independent classifier (ACC_E), and pairwise LPIPS diversity within each prompt [46, 47]. Table 4 shows that BRIDGETWIST attains the best prompt-image alignment and highest diversity simultaneously. Figure 4 shows representative samples from BRIDGETWIST.

Table 4: Text-prompted image generation on CelebA-HQ at $K = 25$. Subscripts give standard deviations; bold marks the best value.

Method	$\text{CLIP}_L \uparrow$	$\text{ACC}_E \uparrow$	LPIPS div. $\uparrow$
FMPS	$0.307_{\pm 0.001}$	$0.689_{\pm 0.020}$	$0.012_{\pm 0.003}$
TFTF	$0.296_{\pm 0.002}$	$0.803_{\pm 0.042}$	$0.501_{\pm 0.032}$
DPS-SB	$0.306_{\pm 0.002}$	$0.794_{\pm 0.024}$	$0.599_{\pm 0.054}$
BRIDGETWIST	$\mathbf{0.311}_{\pm 0.002}$	$\mathbf{0.827}_{\pm 0.044}$	$\mathbf{0.613}_{\pm 0.048}$



"a smiling woman with blonde hair and bangs" "a man with black hair and an exposed forehead" "a woman with a serious expression"

Figure 4: Text-prompted samples.

Super-resolution, deblurring, and inpainting. We further compare with recent image restoration solvers, including PnP-Flow [10], Flower [48], and FlowDPS [49], on CelebA-HQ $4\times$ super-resolution and Gaussian deblurring, and on CIFAR-10 inpainting with half-image and random 50% masks. BRIDGETWIST achieves the highest peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) on deblurring and ranks a close second to PnP-Flow on super-resolution (Table 5). On CIFAR-10, it also achieves the highest SSIM under both inpainting masks. These results show that BRIDGETWIST is competitive with specialized image restoration solvers while supporting more general conditioning. See Appendices D.3 and D.4 for additional results.

Table 5: Image restoration on CelebA-HQ and CIFAR-10. Subscripts give standard deviations; dashes indicate unavailable results; bold marks the best value.

	CelebA-HQ				CIFAR-10	
	Super-resolution		Deblurring		Inpainting	
Method	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	SSIM (half) $\uparrow$	SSIM (random) $\uparrow$
PnP-Flow	$\mathbf{28.47}_{\pm 1.05}$	$\mathbf{0.843}_{\pm 0.020}$	$30.64_{\pm 1.08}$	$0.818_{\pm 0.019}$	$0.623_{\pm 0.104}$	$0.940_{\pm 0.109}$
Flower	$27.88_{\pm 0.99}$	$0.827_{\pm 0.019}$	$30.91_{\pm 1.07}$	$0.839_{\pm 0.015}$	–	–
FlowDPS	$26.43_{\pm 1.86}$	$0.795_{\pm 0.025}$	$29.12_{\pm 0.94}$	$0.819_{\pm 0.028}$	–	–
FMPS	$23.68_{\pm 1.42}$	$0.622_{\pm 0.042}$	$17.92_{\pm 1.67}$	$0.689_{\pm 0.038}$	$0.240_{\pm 0.114}$	$0.554_{\pm 0.106}$
TFTF	$23.78_{\pm 0.95}$	$0.772_{\pm 0.032}$	$27.98_{\pm 0.83}$	$0.737_{\pm 0.032}$	$0.532_{\pm 0.071}$	$0.875_{\pm 0.054}$
DPS-SB	$28.06_{\pm 0.97}$	$0.791_{\pm 0.020}$	$29.49_{\pm 0.82}$	$0.708_{\pm 0.012}$	$0.518_{\pm 0.086}$	$0.782_{\pm 0.082}$
BRIDGETWIST	$28.33_{\pm 0.88}$	$0.836_{\pm 0.024}$	$\mathbf{31.13}_{\pm 0.97}$	$\mathbf{0.847}_{\pm 0.020}$	$\mathbf{0.655}_{\pm 0.072}$	$\mathbf{0.943}_{\pm 0.087}$

4.3 Ablation studies

Table 6 compares endpoint predictor channels under the same SB prior on CIFAR-10. Replacing the full predictor with either the velocity-only Euler predictor or the score-only Tweedie predictor from Section 3.1 substantially degrades both class alignment and sample quality. See Appendices D.5 and D.6 for more diagnostics.

Table 6: CIFAR-10 predictor ablation at $K = 16$. Bold marks the best value in each column.

Predictor	$\text{ACC}_L \uparrow$	$\text{ACC}_E \uparrow$	KID $\downarrow$
Velocity-only	0.526	0.485	0.2805
Score-only	0.711	0.514	0.2339
BRIDGETWIST	**0.948**	**0.914**	**0.0873**

Table 7 isolates the h-transform guidance, tempering, and importance weight correction under the same SB prior on MNIST and CIFAR-10. IS weights and resamples particles only at the endpoint, whereas DPS-SB uses guidance without importance correction. On CIFAR-10, both yield lower ACC_E and higher KID than BRIDGETWIST, as do variants without h-transform guidance or tempering. On MNIST, removing tempering slightly increases ACC_E while worsening KID.

Table 8 varies the tempering exponent $a \in \{0, 1, 2, 3\}$ in $\beta(t) = t^a$ and compares bridge-shaped, constant, and cosine diffusion schedules $g(t)$ on CIFAR-10. The default combination of linear tempering ($a = 1$) and the bridge-shaped schedule performs the best. Tempering is important, since removing it ($a = 0$) largely degrades performance, whereas BRIDGETWIST is insensitive to the specific choice among $a \in \{1, 2, 3\}$ For each a, our bridge-shaped design outperforms the common constant and cosine schedules. Appendix D.7 evaluates tempering in TDS and TFTF.

In summary, these ablations under the same SB prior support attributing the gains to the velocity–score endpoint predictor, h-transform guidance, tempering, and bridge-shaped diffusion. Appendix C.2 further shows the gains of BRIDGETWIST do not come from a better prior model.

Table 7: Sampler omponent ablation at $K = 64$ under the same SB prior. Bold marks the best value in each column.

Variant	h-transform	tempering	weight corr.	MNIST ACC$_E$ ↑	MNIST KID ↓	CIFAR-10 ACC$_E$ ↑	CIFAR-10 KID ↓
IS	×	×	endpoint only	0.920	0.0007	0.865	0.0744
DPS-SB	✓	×	×	0.619	0.0010	0.585	0.1051
BRIDGETWIST w/o h-transform	×	✓	✓	0.932	**0.0003**	0.925	0.0437
BRIDGETWIST w/o tempering	✓	×	✓	**0.958**	0.0004	0.938	0.0813
BRIDGETWIST	✓	✓	✓	0.952	**0.0003**	**0.979**	**0.0403**

Table 8: Joint tempering and diffusion ablation on CIFAR-10 at $K = 64$. Each entry reports ACC$_E$ (↑) / KID (↓). Bold marks the best values.

$a \setminus g(t)$	bridge-shaped (default)	constant ($c_g\sigma$)	cosine ($c_g\sigma\cos(\pi t/2)$)
$a = 0$ (no tempering)	0.938 / 0.0813	0.932 / 0.1489	0.897 / 0.0920
$a = 1$ (default)	**0.979 / 0.0403**	0.967 / 0.1099	0.945 / 0.0534
$a = 2$	0.971 / 0.0426	0.970 / 0.1034	0.916 / 0.0487
$a = 3$	0.967 / 0.0470	0.946 / 0.1013	0.909 / 0.0493

5 Discussion

We introduce BRIDGETWIST, a training-free conditional sampler that reuses pretrained unconditional Schrödinger bridge models. Central to our approach are three contributions to SB conditioning: a plug-in lookahead built from the velocity–score endpoint predictor, temporal tempering that moderates this lookahead when conditional endpoint uncertainty is large, and a bridge-shaped diffusion schedule that concentrates exploration in the path interior. These designs guide and reweight particles through a sequence of intermediate targets toward the endpoint posterior. We establish a finite-particle error bound for estimates under the posterior, and demonstrate competitive performance across class-conditional and text-prompted image generation, and image restoration.

Beyond the sampler itself, the endpoint predictor offers a two-channel view that connects the transport perspective of flow models with the denoising perspective of diffusion models. For an SB under a Brownian reference, the velocity channel describes transport along the bridge, while the score channel corrects for noise in the current state. Our ablations support combining both channels to improve class alignment and sample quality.

Limitations and future work. First, BRIDGETWIST currently is restricted to Schrödinger bridges with the Brownian reference, and extending it to other references require reference-specific endpoint predictors. Second, its twist approximation uses only the conditional endpoint mean and can be inaccurate for strongly curved likelihoods or multimodal conditional endpoint laws. The second-order extension in Appendix D.8 improves some low-dimensional cases, but full covariance estimation is expensive in high dimensions and requires scalable approximations. Third, the sampler targets the posterior induced by the learned and discretized prior, leaving prior-learning and discretization errors. Fourth, preparing the SB prior also requires an additional score head, with a one-time computational overhead relative to directly using pretrained flow or diffusion models. Finally, future work also includes protein design and broader single-cell transport applications, building on the initial example in Appendix D.9.

Acknowledgments and Disclosures

The authors thank the anonymous referees for their insightful and constructive comments, which have led to an improved version of the paper. This work was supported by the China Postdoctoral Science Foundation (Grant No. 2025M783111) and the Shenzhen Science and Technology Program (Grant No. AI2026019). The authors declare no competing interests.

References

[1] Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pages 31372–31403. Curran Associates, Inc., 2023.

[2] Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[3] Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[4] James Matthew Young and O. Deniz Akyildiz. On diffusion posterior sampling via sequential Monte Carlo for zero-shot scaffolding of protein motifs. *Transactions on Machine Learning Research*, 2025.

[5] Qianqian Qu and Jun S. Liu. TFTF: Training-free targeted flow for conditional sampling. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 102577–102606. PMLR, 2026.

[6] Kaiyu Song, Hanjiang Lai, Yan Pan, Kun Yue, and Jian Yin. Flow-matching posterior sampling: A training-free conditional generation for flow matching. *IEEE Transactions on Image Processing*, 35:5970–5981, 2026.

[7] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, volume 35, pages 23593–23606. Curran Associates, Inc., 2022.

[8] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[9] Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training free guided flow-Matching with optimal control. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[10] Ségolène Tiffany Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. PnP-flow: Plug-and-Play image restoration with flow matching. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[11] Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi S. Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[12] Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte Carlo guided Denoising Diffusion models for Bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[13] Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55906–55949. PMLR, 2025.

[14] Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55810–55827. PMLR, 2025.

[15] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 17695–17709. Curran Associates, Inc., 2021.

[16] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 36, pages 62183–62223. Curran Associates, Inc., 2023.

[17] Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Adversarial Schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 37, pages 89612–89651. Curran Associates, Inc., 2024. doi: 10.52202/079017-2845.

[18] Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free Schrödinger bridges via score and flow matching. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1279–1287. PMLR, 2024.

[19] Tianrong Chen, Guan-Horng Liu, and Evangelos A. Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations (ICLR)*, 2022.

[20] Eslam Abdelrahman, Liangbing Zhao, Vincent Tao Hu, Matthieu Cord, Patrick Perez, and Mohamed Elhoseiny. ToddlerDiffusion: Interactive structured image generation with cascaded Schrödinger bridge. In *International Conference on Learning Representations (ICLR)*, 2025.

[21] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-image Schrödinger Bridge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22042–22062. PMLR, 2023.

[22] Jingjun Yi, Qi Bi, Hao Zheng, Huimin Huang, Yixian Shen, Haolan Zhan, Wei Ji, Yawen Huang, Yuexiang Li, Xian Wu, and Yefeng Zheng. Degradation-aware dynamic Schrödinger bridge for unpaired image restoration. In *Advances in Neural Information Processing Systems*, volume 38, pages 2415–2439. Curran Associates, Inc., 2025.

[23] Clémence Fournié, Elias Ventre, Ulysse Herbach, Aymeric Baradat, Olivier Gandrillon, and Fabien Crauste. Cell trajectory inference based on Schrödinger problem and a mechanistic model of stochastic gene expression. *npj Systems Biology and Applications*, 12(1):120, 2026.

[24] Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Conditional simulation using diffusion Schrödinger bridges. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1792–1802. PMLR, 2022.

[25] Hanwen Huang and Manyu Huang. Schrödinger bridge based deep conditional generative learning. *Journal of Multivariate Analysis*, 210:105486, 2025. doi: 10.1016/j.jmva.2025.105486.

[26] Chang Li, Zehua Chen, Fan Bao, and Jun Zhu. Bridge-SR: Schrödinger Bridge for efficient SR. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. doi: 10.1109/ICASSP49660.2025.10890104.

[27] Ming Lin, Rong Chen, and Per Mykland. On generating Monte Carlo samples of continuous diffusion bridges. *Journal of the American Statistical Association*, 105(490):820–838, 2010.

[28] Ming Lin, Rong Chen, and Jun S. Liu. Lookahead strategies for sequential Monte Carlo. *Statistical Science*, 28(1):69–94, 2013.

[29] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025.

[30] Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'Institut Henri Poincaré*, 2(4):269–310, 1932.

[31] Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014.

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth International Conference on Learning Representations (ICLR)*, 2021.

[33] Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[34] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[35] Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

[36] Pierre Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, 2004.

[37] Nick Whiteley and Anthony Lee. Twisted particle filters. *The Annals of Statistics*, 42(1): 115–141, 2014.

[38] Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

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

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[41] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[42] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[43] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018.

[44] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185. PMLR, 2020.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[47] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[48] Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A flow-matching solver for inverse problems. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[49] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12328–12337, 2025.

[50] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[52] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[53] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[54] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[55] Yanbo Xu, Yueqin Yin, Liming Jiang, Qianyi Wu, Chengyao Zheng, Chen Change Loy, Bo Dai, and Wayne Wu. TransEditor: Transformer-based dual-space GAN for highly controllable facial editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7683–7692, 2022.

[56] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019.

[57] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019.

Appendix contents

A Proofs **16**

A.1 Proof of Proposition 1 16

A.2 Inference SDE preserves the SB marginal 17

A.3 Proof of Theorem 1 18

A.4 Proof of Remark 1 23

B Pretraining details **23**

C Implementation details **24**

C.1 Two-dimensional toy benchmark 24

C.2 Unconditional models and likelihoods 25

C.3 Hyperparameters and metrics 26

D Additional results **27**

D.1 Computation cost 27

D.2 Joint particle and time-step allocation 27

D.3 Qualitative samples 27

D.4 Expanded CIFAR-10 inpainting comparison 28

D.5 Calibration of the endpoint predictor 28

D.6 SMC diagnostics 30

D.7 Tempering in TDS and TFTF 31

D.8 Second-order twist extension 31

D.9 Conditional transition with a non-Gaussian source 32

A Proofs

In this appendix, we prove main theoretical results. Appendix A.1 proves the endpoint decomposition in Proposition 1, Appendix A.2 proves the marginal preservation property of the inference SDE, and Appendix A.3 proves the finite-particle bound of Theorem 1; Appendix A.4 proves Remark 1.

We work in the Brownian-reference Schrödinger bridge setup of Section 2.1: conditional on (X_0, X_1),

$$X_t \mid (X_0, X_1) \sim \mathcal{N}(M_t, \tau_t^2 I), \qquad M_t = (1-t)X_0 + tX_1, \qquad \tau_t^2 = \sigma^2 t(1-t).$$

We also write $D = X_1 - X_0$. Throughout, p_t denotes the SB marginal density and is assumed to be continuously differentiable on $\mathbb{R}^d$ for $t \in (0,1)$.

A.1 Proof of Proposition 1

Proof. Fix $t \in (0,1)$ and $x \in \mathbb{R}^d$. The Brownian reference SB is a mixture of Brownian bridges over its endpoint coupling. Conditional on $X_t = x$ and $X_1 = x_1$, the future path is therefore a Brownian bridge from (t,x) to $(1,x_1)$, satisfying

$$\mathrm{d}X_s = \frac{x_1 - X_s}{1-s}\,\mathrm{d}s + \sigma\,\mathrm{d}\widetilde{W}_s, \qquad t \le s < 1,$$

where $\widetilde{W}$ is Brownian motion under the conditional law. Averaging this drift at time t over $X_1 | X_t = x$ gives

$$b_t^\star(x) = \mathbb{E}\Big[\frac{X_1 - X_t}{1-t}\,\Big|\, X_t = x\Big] = \frac{\mathbb{E}[X_1 \mid X_t = x] - x}{1-t}.$$

Using $b_t^\star = v_t^\star + \sigma^2 s_t^\star/2$ and rearranging gives

$$\mathbb{E}[X_1 \mid X_t = x] = x + (1-t)v_t^\star(x) + \frac{\sigma^2}{2}(1-t)s_t^\star(x). \tag{13}$$

To recover $\mathbb{E}[M_t | X_t = x]$, let ν_t be the law of M_t and ϕ_{τ_t} the $\mathcal{N}(0, \tau_t^2 I)$ density. Since $X_t | M_t \sim \mathcal{N}(M_t, \tau_t^2 I)$, the marginal density of X_t is the convolution

$$p_t(x) = \int \phi_{\tau_t}(x-m)\,\nu_t(\mathrm{d}m). \tag{14}$$

Differentiating Eq. (14) with respect to x gives

$$\begin{aligned}
\nabla_x p_t(x) &= \int \nabla_x \phi_{\tau_t}(x-m)\,\nu_t(\mathrm{d}m) \\
&= -\frac{1}{\tau_t^2}\int (x-m)\phi_{\tau_t}(x-m)\,\nu_t(\mathrm{d}m) \\
&= -\frac{1}{\tau_t^2}\Big[x p_t(x) - \int m\phi_{\tau_t}(x-m)\,\nu_t(\mathrm{d}m)\Big].
\end{aligned} \tag{15}$$

By Bayes' rule, the conditional law of M_t given $X_t = x$ is $\phi_{\tau_t}(x-m)\nu_t(\mathrm{d}m)/p_t(x)$, so

$$\int m\phi_{\tau_t}(x-m)\,\nu_t(\mathrm{d}m) = p_t(x)\mathbb{E}[M_t \mid X_t = x].$$

Dividing Eq. (15) by $p_t(x)$ and rearranging yields Tweedie's identity for the interpolation M_t,

$$\mathbb{E}[M_t \mid X_t = x] = x + \tau_t^2 \nabla_x \log p_t(x) = x + \sigma^2 t(1-t)s_t^\star(x). \tag{16}$$

Finally, $X_1 = M_t + (1-t)D$ and Eqs. (13) and (16) yield

$$\mathbb{E}[D \mid X_t = x] = \frac{\mathbb{E}[X_1 \mid X_t = x] - \mathbb{E}[M_t \mid X_t = x]}{1-t} = v_t^\star(x) + \frac{1-2t}{2}\sigma^2 s_t^\star(x). \tag{17}$$

□

Corollary 1 (Source-side conditional mean)**.** *Under the same assumptions, for every $t \in (0,1)$ and $x \in \mathbb{R}^d$,*

$$\mathbb{E}[X_0 \mid X_t = x] = x - t\, v_t^\star(x) + \frac{\sigma^2}{2}\, t\, s_t^\star(x).$$

Proof. The source state satisfies $X_0 = M_t - tD$. Combining (16) and (17),

$$\begin{aligned}
\mathbb{E}[X_0 \mid X_t = x] &= \mathbb{E}[M_t \mid X_t = x] - t\,\mathbb{E}[D \mid X_t = x] \\
&= x + \sigma^2 t(1-t) s_t^\star(x) - t v_t^\star(x) - t\frac{1-2t}{2}\sigma^2 s_t^\star(x) \\
&= x - t v_t^\star(x) + \sigma^2 t\left[(1-t) - \frac{1-2t}{2}\right] s_t^\star(x) \\
&= x - t v_t^\star(x) + \frac{\sigma^2}{2} t s_t^\star(x).
\end{aligned}$$

□

A.2 Inference SDE preserves the SB marginal

Proposition 2. *Let $g \in C([0,1], \mathbb{R}_{\geq 0})$ be an inference diffusion schedule. The Brownian-reference Schrodinger bridge SDE (1) is*

$$\mathrm{d}\bar{X}_t = b_t^\star(\bar{X}_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t, \quad b_t^\star(x) = v_t^\star(x) + \frac{1}{2}\sigma^2 s_t^\star(x), \quad \bar{X}_0 \sim q_0, \tag{18}$$

with $s_t^\star(x) = \nabla \log p_t(x)$ and $q_0 = p_0$. Define the inference-time drift

$$b_{\theta,g}(t,x) := v_t^\star(x) + \frac{1}{2} g(t)^2 s_t^\star(x),$$

and the corresponding inference SDE

$$\mathrm{d}X_t = b_{\theta,g}(t, X_t)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t, \qquad X_0 \sim q_0. \tag{19}$$

For any such g, the marginal density of X_t under (19) equals the SB marginal density p_t of (18) for every $t \in (0,1)$.

Proof. The SB probability-flow ODE associated with the marginal velocity $v^\star$ transports $p_0 = q_0$ along the marginal densities $\{p_t\}_{t\in(0,1)}$. Equivalently, these densities satisfy the continuity equation

$$\partial_t p_t(x) = -\nabla \cdot [p_t(x) v_t^\star(x)]. \tag{20}$$

Let $\tilde{p}_t$ denote the marginal density of the SDE (19). Its Fokker–Planck equation is

$$\partial_t \tilde{p}_t(x) = -\nabla \cdot [\tilde{p}_t(x) b_{\theta,g}(t,x)] + \frac{1}{2} g(t)^2 \Delta \tilde{p}_t(x), \tag{21}$$

with initial condition $\tilde{p}_0 = q_0$. Because the SB source constraint gives $p_0 = q_0$, the two initial conditions coincide. We now check that the SB density p_t also solves (21). Substituting p_t in the right-hand side of (21), and using $b_{\theta,g} = v^\star + \frac{1}{2} g^2 s^\star$, gives

$$\begin{aligned}
&-\nabla \cdot [p_t b_{\theta,g}] + \frac{1}{2} g(t)^2 \Delta p_t \\
&\quad = -\nabla \cdot [p_t v^\star] - \frac{1}{2} g(t)^2 \nabla \cdot [p_t s^\star] + \frac{1}{2} g(t)^2 \Delta p_t.
\end{aligned} \tag{22}$$

Since $s_t^\star(x) = \nabla \log p_t(x)$,

$$p_t(x) s_t^\star(x) = p_t(x) \nabla \log p_t(x) = \nabla p_t(x),$$

and hence

$$\nabla \cdot [p_t s^\star] = \nabla \cdot [\nabla p_t] = \Delta p_t.$$

The last two terms in (22) therefore cancel exactly:

$$-\frac{1}{2} g(t)^2 \nabla \cdot [p_t s^\star] + \frac{1}{2} g(t)^2 \Delta p_t = 0.$$

Then the right-hand side of (22) reduces to $-\nabla \cdot [p_t v^\star]$, which equals $\partial_t p_t$ by the continuity equation (20). Thus p_t satisfies the same Fokker–Planck equation and the same initial condition as $\tilde{p}_t$, and hence $\tilde{p}_t = p_t$ for all $t \in (0, 1)$.

This result is independent of the specific choice of g. Changing g adds both a diffusion term and the drift term $\frac{1}{2} g(t)^2 \nabla \log p_t$ with the same coefficient, so the two contributions cancel at the density level. This is the flexible schedule property used in Section 3.2. □

A.3 Proof of Theorem 1

Theorem 1 uses the following regularity conditions on the tempered twist and the Gaussian proposal kernels of Algorithm 1.

Assumption 1 (Endpoint twist consistency)**.** *For all $x \in \mathbb{R}^d$, $\hat{h}_N(x)^{\beta_N} = L_y(x)$.*

Assumption 2 (Bounded likelihood and twist)**.** *There are constants $0 < \ell_n \le u_n < \infty$ such that $\ell_n \le \hat{h}_n(x)^{\beta_n} \le u_n$ for all $x \in \mathbb{R}^d$ and $n \in \{0, \dots, N\}$; in particular, $\ell_N \le L_y(x) \le u_N$.*

Assumption 3 (Bounded log-twist gradient)**.** *For each n with $\beta_n > 0$, the map $x \mapsto \log \hat{h}_n(x)^{\beta_n}$ is continuously differentiable and has uniformly bounded gradient.*

Assumption 4 (Proposal coverage)**.** *For every n, the prior covariance is nondegenerate, i.e., $\sigma^2_{M,n} := g(t_{n-1})^2 \Delta_n > 0$, and the proposal covariance $\Sigma_{q,n}$ of Section 3.2 strictly dominates it:*

$$\Sigma_{M,n} = \sigma^2_{M,n} I, \qquad \Sigma_{q,n} = (\sigma^2_{M,n} + \varepsilon) I, \qquad \varepsilon > 0.$$

The nondegeneracy condition is satisfied by the bridge-shaped schedule with $\delta_g > 0$ used in the experiments and by any strictly positive inference diffusion schedule on the discretization grid.

Mildness of the assumptions. Assumption 1 holds by construction because $\beta_N = 1$ and $\hat{h}_N(x) = L_y(x)$. Assumption 2 is satisfied by bounded likelihoods with a positive floor and by CLIP likelihoods with bounded similarity scores and a finite guidance scale λ. It also implies $\ell_N \le h_n(x) \le u_N$, so the relative twist mismatch $R_{n,N}$ in Theorem 1 is finite. Assumption 3 follows from bounded likelihood gradients and bounded Jacobians of (v_θ, s_θ) through the chain rule applied to Eq. (8), matching the regularity used in diffusion twisting analyses [1, Appendix A.5]; the explicit chain-rule derivation is recorded below. Assumption 4 is the standard absolute-continuity safeguard: the small εI inflation gives the proposal heavier Gaussian tails than the prior kernel.

Lemma 1 (Ratio estimator)**.** *Let $(Z_i, F_i)_{i=1}^K$ be independent pairs with $Z_i > 0$, $\mathbb{E}[Z_i^2] < \infty$, and $F_i \in [m, M]$ almost surely. For $\theta := \sum_i \mathbb{E}[Z_i F_i] / \sum_i \mathbb{E}[Z_i]$ and $\hat{\theta} := \sum_i Z_i F_i / \sum_i Z_i$, we have*

$$\mathbb{E}[(\hat{\theta} - \theta)^2] \le 8(M - m)^2 \, \frac{\sum_i \mathbb{E}[Z_i^2]}{(\sum_i \mathbb{E}[Z_i])^2}.$$

Proof. Set $D := \sum_i Z_i$, $d := \mathbb{E}[D]$, and $Y := \sum_i Z_i (F_i - \theta)$. Independence gives $\mathbb{E}[Y] = 0$, $\mathbb{E}[Y^2] \le (M - m)^2 \sum_i \mathbb{E}[Z_i^2]$, and $\mathbb{P}(D < d/2) \le 4 \sum_i \mathbb{E}[Z_i^2] / d^2$ by Chebyshev's inequality. Since $\hat{\theta} - \theta = Y / D$ and $|\hat{\theta} - \theta| \le M - m$, splitting on $D \ge d/2$ yields

$$\mathbb{E}[(\hat{\theta} - \theta)^2] \le \frac{4 \mathbb{E}[Y^2]}{d^2} + (M - m)^2 \mathbb{P}(D < d/2) \le \frac{8(M - m)^2 \sum_i \mathbb{E}[Z_i^2]}{d^2}.$$

□

Proof of Theorem 1. The proof has five steps and follows the standard twisted SMC theory used in prior work [1] and related Feynman–Kac analyses [35, Chapters 5 and 11]. We first verify the final Feynman–Kac target on path space. We then verify that the incremental weights are uniformly bounded. We next express Algorithm 1 as a Feynman–Kac particle system whose future kernels are the likelihood-conditioned prior dynamics, bound the sampling error introduced at one step, and propagate the one-step errors to step $n = N$.

Step 1: verify the Feynman–Kac target and endpoint marginal. Let Q_N denote the proposal path law used by Algorithm 1,

$$Q_N(\mathrm{d}x_{t_{0:N}}) = q_0(\mathrm{d}x_{t_0}) \prod_{n=1}^{N} q_n(x_{t_n} \mid x_{t_{n-1}}, y)\, \mathrm{d}x_{t_n},$$

where q_0 is the SB source distribution; the algorithm initializes particles directly from q_0, which is also the initial marginal of the discretized SB prior path measure $P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}}) = q_0(\mathrm{d}x_{t_0}) \prod_{n=1}^{N} M_n(x_{t_{n-1}}, \mathrm{d}x_{t_n})$ defined in Section 3.2. The algorithm attaches the initial weight $\hat{h}_0(x_{t_0})^{\beta_0}$ and then the incremental weights (11). Therefore the unnormalized path measure represented by the weighted particles at step N is

$$\Gamma^{\beta}_N(\mathrm{d}x_{t_{0:N}}) = Q_N(\mathrm{d}x_{t_{0:N}}) \hat{h}_0(x_{t_0})^{\beta_0} \prod_{n=1}^{N} \omega_n(x_{t_{n-1}}, x_{t_n}). \tag{23}$$

Substituting the definition of ω_n gives the telescoping product

$$\begin{aligned} \hat{h}_0(x_{t_0})^{\beta_0} \prod_{n=1}^{N} \omega_n(x_{t_{n-1}}, x_{t_n}) &= \hat{h}_0(x_{t_0})^{\beta_0} \prod_{n=1}^{N} \frac{M_n(x_{t_{n-1}}, x_{t_n}) \hat{h}_n(x_{t_n})^{\beta_n}}{q_n(x_{t_n} \mid x_{t_{n-1}}, y) \hat{h}_{n-1}(x_{t_{n-1}})^{\beta_{n-1}}} \\ &= \frac{\prod_{n=1}^{N} M_n(x_{t_{n-1}}, x_{t_n})}{\prod_{n=1}^{N} q_n(x_{t_n} \mid x_{t_{n-1}}, y)} \hat{h}_N(x_{t_N})^{\beta_N}. \end{aligned} \tag{24}$$

All intermediate twist powers cancel pairwise, and only the endpoint twist remains. Multiplying (24) by the proposal path measure in (23) yields

$$\Gamma^{\beta}_N(\mathrm{d}x_{t_{0:N}}) = q_0(\mathrm{d}x_{t_0}) \prod_{n=1}^{N} M_n(x_{t_{n-1}}, \mathrm{d}x_{t_n})\, \hat{h}_N(x_{t_N})^{\beta_N} = P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}})\, \hat{h}_N(x_{t_N})^{\beta_N},$$

where the second equality uses the factorization of $P^{(N)}_{\theta,g}$ recorded above. By Assumption 1, $\hat{h}_N(x_{t_N})^{\beta_N} = L_y(x_{t_N})$, and hence

$$\Gamma^{\beta}_N(\mathrm{d}x_{t_{0:N}}) = P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}})\, L_y(x_{t_N}). \tag{25}$$

Thus the final Feynman–Kac target is the discretized SB prior path law tilted only by the true endpoint likelihood. The intermediate twists and the tempering schedule affect the proposal and weight variance, but not the final target. Normalizing this path measure also identifies its endpoint marginal. Let $Z^{\beta}_N = \Gamma^{\beta}_N(1)$ and $\pi^{\beta}_N = \Gamma^{\beta}_N / Z^{\beta}_N$. For any bounded measurable $\varphi : \mathbb{R}^d \to \mathbb{R}$, (25) gives

$$\begin{aligned} \int \varphi(x_{t_N})\, \pi^{\beta}_N(\mathrm{d}x_{t_{0:N}}) &= \frac{\int \varphi(x_{t_N})\, L_y(x_{t_N})\, P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}})}{\int L_y(x_{t_N})\, P^{(N)}_{\theta,g}(\mathrm{d}x_{t_{0:N}})} \\ &= \frac{\int \varphi(x_{t_N})\, L_y(x_{t_N})\, p^{(N)}_{\theta}(\mathrm{d}x_{t_N})}{\int L_y(x_{t_N})\, p^{(N)}_{\theta}(\mathrm{d}x_{t_N})} \\ &= \mathbb{E}_{X_{t_N} \sim \pi^{\star,(N)}}[\varphi(X_{t_N})]. \end{aligned} \tag{26}$$

The second equality integrates out $x_{t_{0:N-1}}$ under the discretized SB prior path law, whose endpoint marginal is $p^{(N)}_{\theta}$. This proves that the endpoint marginal of π^{β}_N is exactly the posterior $\pi^{\star,(N)}$ appearing in Theorem 1.

Step 2: bound the incremental weights. The incremental weight factors as

$$\omega_n(x_{t_{n-1}}, x_{t_n}) = \frac{\hat{h}_n(x_{t_n})^{\beta_n}}{\hat{h}_{n-1}(x_{t_{n-1}})^{\beta_{n-1}}} \frac{M_n(x_{t_{n-1}}, x_{t_n})}{q_n(x_{t_n} \mid x_{t_{n-1}}, y)}.$$

By Assumption 2, the first factor lies in $[\ell_n/u_{n-1}, u_n/\ell_{n-1}]$ and is uniformly bounded on the finite grid. It remains to bound the Gaussian density ratio.

Write

$$M_n(\cdot \mid x_{t_{n-1}}) = \mathcal{N}(m_{M,n}, a_n I), \qquad q_n(\cdot \mid x_{t_{n-1}}, y) = \mathcal{N}(m_{q,n}, c_n I),$$

where $a_n := g(t_{n-1})^2 \Delta_n > 0$, $c_n := a_n + \varepsilon$, and $\delta_n := m_{q,n} - m_{M,n}$. By Eqs. (9) and (10),

$$\begin{aligned}\delta_n(x_{t_{n-1}}) &= w_g\, g(t_{n-1})^2 \Delta_n\, \beta_{n-1} \nabla_x \log \hat{h}_{n-1}(x_{t_{n-1}}) \\ &= w_g\, g(t_{n-1})^2 \Delta_n\, \nabla_x \log\big(\hat{h}_{n-1}(x_{t_{n-1}})^{\beta_{n-1}}\big).\end{aligned}$$

If $\beta_{n-1} = 0$ or $w_g = 0$, this shift is zero; otherwise Assumption 3, together with continuity of g on $[0,1]$, the finite hyperparameter $w_g \geq 0$, and the finite grid, implies that $\|\delta_n(x_{t_{n-1}})\|$ is uniformly bounded over $(n, x_{t_{n-1}})$. For $z := x_{t_n} - m_{M,n}$,

$$\begin{aligned}\log \frac{M_n(x_{t_{n-1}}, x_{t_n})}{q_n(x_{t_n} \mid x_{t_{n-1}}, y)} &= \frac{d}{2} \log \frac{c_n}{a_n} - \frac{1}{2a_n}\|z\|^2 + \frac{1}{2c_n}\|z - \delta_n\|^2 \\ &= \frac{d}{2} \log \frac{c_n}{a_n} - \frac{c_n - a_n}{2a_n c_n}\|z\|^2 - \frac{1}{c_n}\langle z, \delta_n\rangle + \frac{1}{2c_n}\|\delta_n\|^2 \\ &\leq \frac{d}{2} \log \frac{c_n}{a_n} - \frac{\varepsilon}{2a_n c_n}\|z\|^2 + \frac{1}{c_n}\|z\|\,\|\delta_n\| + \frac{1}{2c_n}\|\delta_n\|^2. \qquad (27)\end{aligned}$$

For any $r \geq 0$,

$$-\frac{\varepsilon}{2a_n c_n} r^2 + \frac{\|\delta_n\|}{c_n} r \leq \frac{a_n}{2c_n \varepsilon}\|\delta_n\|^2,$$

which is the maximum of the scalar quadratic in r. Applying this bound to $r = \|z\|$ in (27) gives

$$\log \frac{M_n(x_{t_{n-1}}, x_{t_n})}{q_n(x_{t_n} \mid x_{t_{n-1}}, y)} \leq \frac{d}{2} \log \frac{c_n}{a_n} + \frac{\|\delta_n\|^2}{2\varepsilon}.$$

Because the time grid is finite, Assumption 4 makes the determinant term $\frac{d}{2}\log(c_n/a_n)$ finite uniformly over n. The bound on δ_n derived from Assumption 3 makes the second term uniformly finite. Therefore M_n/q_n is uniformly bounded over $(n, x_{t_{n-1}}, x_{t_n})$. Combining this with Assumption 2, each ω_n is positive and uniformly bounded. The initial weight $\hat{h}_0^{\beta_0}$ is also bounded by Assumption 2.

Step 3: describe the exact distribution updates. We next write the exact updates that the particles approximate, so that we can isolate the error introduced at each step and follow its effect on the endpoint. Write $H_n := \hat{h}_n^{\beta_n}$. Starting from a state x at step $n-1$, propagation and weighting give the following expected contribution of a bounded test function f,

$$Q_n f(x) := \int q_n(x' \mid x, y)\omega_n(x, x') f(x')\,\mathrm{d}x' = \frac{1}{H_{n-1}(x)} \int M_n(x, \mathrm{d}x') H_n(x') f(x').$$

The equality follows by substituting the incremental weight, which cancels the proposal density. For a probability law μ at step $n-1$, write $\mu(f) := \int f(x)\,\mu(\mathrm{d}x)$. Averaging these contributions and dividing by the total weight gives the exact updated expectation

$$\Phi_n(\mu)(f) := \frac{\mu(Q_n f)}{\mu(Q_n \mathbf{1})}.$$

Since $\beta_0 = 0$ and $H_0 \equiv 1$, the exact intermediate marginals start at $\eta_0 = q_0$ and satisfy $\eta_n = \Phi_n(\eta_{n-1})$. Step 1 identifies their terminal marginal as $\eta_N = \pi^{\star,(N)}$.

To describe the remaining updates from step p to step n, let $M_{p+1:n}$ denote the composition of the prior kernels over those steps. Repeated propagation and weighting compose the operators $Q_{p+1}, \ldots, Q_n$. As in Step 1, the intermediate twist factors cancel, giving

$$Q_{p,n} f(x) := (Q_{p+1} \cdots Q_n f)(x) = \frac{1}{H_p(x)} \int M_{p+1:n}(x, \mathrm{d}z) H_n(z) f(z).$$

Thus the exact continuation of a law μ from step p to step n is

$$\Phi_{p,n}(\mu)(f) := \frac{\mu(Q_{p,n} f)}{\mu(Q_{p,n} \mathbf{1})},$$

with $Q_{n,n} = I$ and $\Phi_{n,n}(\mu) = \mu$ when no updates remain. In particular, $\Phi_{n-1,n} = \Phi_n$.

At the endpoint, $H_N = L_y$ and $h_p(x) = \int M_{p+1:N}(x, \mathrm{d}z) L_y(z)$. Normalizing this likelihood-weighted prior transition gives the posterior transition kernel

$$P^y_{p,N}(x, \mathrm{d}z) = \frac{M_{p+1:N}(x, \mathrm{d}z) L_y(z)}{h_p(x)}.$$

Consequently, the remaining weighted contribution factors as

$$Q_{p,N} f(x) = \frac{1}{H_p(x)} \int M_{p+1:N}(x, \mathrm{d}z) L_y(z) f(z) = \frac{h_p(x)}{H_p(x)} P^y_{p,N} f(x),$$

where $P^y_{p,N} f(x)$ is the conditional expectation of the endpoint function f under this kernel. Define the total remaining weight by $u_{p,N} := Q_{p,N}\mathbf{1} = h_p / H_p$. The exact continuation therefore first reweights the current law by $u_{p,N}$ and then applies the posterior transition,

$$\Phi_{p,N}(\mu)(f) = \frac{\mu(u_{p,N} P^y_{p,N} f)}{\mu(u_{p,N})}.$$

Here $u_{N,N} \equiv 1$ and $P^y_{N,N} = I$. Assumption 2 gives $\ell_N / u_p \le u_{p,N}(x) \le u_N / \ell_p$, so its relative variation is $\sup u_{p,N} / \inf u_{p,N} = R_{p,N} < \infty$. This factorization separates the weight correction and the posterior transition whose effects on error we bound in Step 5.

Step 4: bound the error introduced at one step. We now compare one particle update with the exact update from the same starting particles. Write the weighted particle estimate of a test function as $\eta^K_n(f) := \sum_{i=1}^K W^{(i)}_n f(X^{(i)}_{t_n})$. For $n \ge 1$, define its comparison law by

$$\tilde{\eta}^K_n := \Phi_n(\eta^K_{n-1}), \qquad \tilde{\eta}^K_0 := \eta_0.$$

The difference $\eta^K_n - \tilde{\eta}^K_n$ isolates the sampling error added at step n while holding the previous particle approximation fixed. Let $\mathcal{F}^K_{n-1}$ be the sigma-field generated by the particle history through step $n-1$; this history determines the ESS resampling decision. For every bounded $\mathcal{F}^K_{n-1}$-measurable test function f, we will show

$$\mathbb{E}\Big[\big| (\eta^K_n - \tilde{\eta}^K_n)(f) \big|^2 \,\Big|\, \mathcal{F}^K_{n-1} \Big]^{1/2} \le \frac{\alpha_n}{\sqrt{K}} \operatorname{osc}(f), \qquad n \in \{0, \dots, N\}, \tag{28}$$

where α_n is deterministic and independent of K. At $n = 0$, the bound is unconditional with deterministic f. The independent samples from q_0 have equal weights, so $\mathrm{Var}[\eta^K_0(f)] \le \operatorname{osc}(f)^2/(4K)$ and we can take $\alpha_0 = 1/2$.

For $n \ge 1$, the first two conditional moments of an incremental weight are

$$m_n(x) := \int q_n(x' \mid x, y) \omega_n(x, x') \, \mathrm{d}x' = Q_n \mathbf{1}(x), \qquad v_n(x) := \int q_n(x' \mid x, y) \omega_n(x, x')^2 \, \mathrm{d}x'.$$

Assumption 2 bounds the mean weight, and Step 2 bounds each realized weight by a finite constant $\bar{\omega}_n$. These bounds imply

$$\underline{m}_n := \frac{\ell_n}{u_{n-1}} \le m_n(x) \le \frac{u_n}{\ell_{n-1}} =: \overline{m}_n, \qquad v_n(x) \le \bar{\omega}_n m_n(x) \le \kappa_n m_n(x)^2,$$

where $\kappa_n := \bar{\omega}_n / \underline{m}_n$ and $b_n := \overline{m}_n / \underline{m}_n$. We apply Lemma 1 with $F_i := f(X^{(i)}_{t_n})$, choosing Z_i according to the resampling decision.

If resampling occurs, the reset weights are equal, so we can take $Z_i := \omega_n(\bar{X}^{(i)}_{t_{n-1}}, X^{(i)}_{t_n})$ and obtain $\hat{\theta} = \eta^K_n(f)$. Stratified resampling uses independent uniforms on the K strata, followed by independent proposal draws, making the pairs (Z_i, F_i) independent conditional on $\mathcal{F}^K_{n-1}$. Its expected ancestor counts satisfy

$$\sum_{i=1}^K \mathbb{P}(A^{(i)}_{n-1} = j \mid \mathcal{F}^K_{n-1}) = K W^{(j)}_{n-1}.$$

Summing over the ancestors and proposal draws therefore gives

$$\theta = \frac{\sum_i \mathbb{E}[Z_i F_i \mid \mathcal{F}_{n-1}^K]}{\sum_i \mathbb{E}[Z_i \mid \mathcal{F}_{n-1}^K]} = \frac{K\eta_{n-1}^K(Q_n f)}{K\eta_{n-1}^K(m_n)} = \tilde{\eta}_n^K(f),$$
$$\frac{\sum_i \mathbb{E}[Z_i^2 \mid \mathcal{F}_{n-1}^K]}{(\sum_i \mathbb{E}[Z_i \mid \mathcal{F}_{n-1}^K])^2} = \frac{\eta_{n-1}^K(v_n)}{K\eta_{n-1}^K(m_n)^2} \le \frac{\kappa_n b_n^2}{K}.$$

If resampling is skipped, we include the previous particle weights by setting

$$Z_i := W_{n-1}^{(i)} \omega_n(X_{t_{n-1}}^{(i)}, X_{t_n}^{(i)}).$$

The proposal draws again give conditionally independent pairs and $\hat{\theta} = \eta_n^K(f)$. In this case,

$$\theta = \frac{\eta_{n-1}^K(Q_n f)}{\eta_{n-1}^K(m_n)} = \tilde{\eta}_n^K(f),$$
$$\frac{\sum_i \mathbb{E}[Z_i^2 \mid \mathcal{F}_{n-1}^K]}{(\sum_i \mathbb{E}[Z_i \mid \mathcal{F}_{n-1}^K])^2} = \frac{\sum_i (W_{n-1}^{(i)})^2 v_n(X_{t_{n-1}}^{(i)})}{\eta_{n-1}^K(m_n)^2} \le \kappa_n b_n^2 \sum_i (W_{n-1}^{(i)})^2 \le \frac{\kappa_n b_n^2}{\rho K}.$$

The last inequality follows because skipping resampling requires $\mathrm{ESS}_{n-1} \ge \rho K$. Since $\rho \le 1$, both cases satisfy the second-moment bound $\kappa_n b_n^2/(\rho K)$. Lemma 1, applied conditionally on the particle history, proves Eq. (28) with $\alpha_n = b_n\sqrt{8\kappa_n/\rho}$.

Step 5: propagate the one-step errors to the endpoint. We finally express the terminal error as a sum of the one-step sampling errors, each followed by the remaining exact updates. For $p \ge 1$, the definition of $\tilde{\eta}_p^K$ gives

$$\Phi_{p,N}(\tilde{\eta}_p^K) = \Phi_{p,N}\big(\Phi_p(\eta_{p-1}^K)\big) = \Phi_{p-1,N}(\eta_{p-1}^K).$$

The second equality combines the exact update from step $p-1$ to step p with the remaining exact updates. When we sum the differences between the particle laws and their comparison laws, these consecutive terms cancel. Using $\Phi_{N,N}(\eta_N^K) = \eta_N^K$ and $\Phi_{0,N}(\eta_0) = \eta_N$, we obtain

$$\eta_N^K - \eta_N = \sum_{p=0}^{N} \Big[\Phi_{p,N}(\eta_p^K) - \Phi_{p,N}(\tilde{\eta}_p^K)\Big].$$

To apply Step 4 to each summand, we rewrite the propagated difference as a one-step error evaluated on a transformed test function. Fix p and write $\mu := \eta_p^K$, $\nu := \tilde{\eta}_p^K$, and $B := \Phi_{p,N}(\nu)(\varphi)$. The factorization in Step 3 gives

$$\Phi_{p,N}(\mu)(\varphi) - \Phi_{p,N}(\nu)(\varphi) = \frac{\mu(u_{p,N} P_{p,N}^y \varphi)}{\mu(u_{p,N})} - B$$
$$= \frac{(\mu-\nu)\big(u_{p,N}(P_{p,N}^y\varphi - B)\big)}{\mu(u_{p,N})},$$

where the last equality uses $\nu\big(u_{p,N}(P_{p,N}^y\varphi - B)\big) = 0$. Define

$$D\varphi(x) := \frac{u_{p,N}(x)}{\nu(u_{p,N})}\big(P_{p,N}^y\varphi(x) - B\big).$$

Then the propagated difference becomes

$$\Phi_{p,N}(\mu)(\varphi) - \Phi_{p,N}(\nu)(\varphi) = \frac{\nu(u_{p,N})}{\mu(u_{p,N})}(\mu-\nu)(D\varphi).$$

Because $\nu = \tilde{\eta}_p^K$ depends only on the particle history through step $p-1$, the function $D\varphi$ is $\mathcal{F}_{p-1}^K$-measurable and can be used in Eq. (28). At $p = 0$, it is deterministic.

It remains to bound the prefactor and the oscillation of this test function. The definition of $R_{p,N}$ gives

$$\frac{\nu(u_{p,N})}{\mu(u_{p,N})} \le \frac{\sup u_{p,N}}{\inf u_{p,N}} = R_{p,N}.$$

Also, B is a weighted average of $P^y_{p,N}\varphi$, so it lies between that function's infimum and supremum. Since $u_{p,N}(x)/\nu(u_{p,N}) \le R_{p,N}$, we have

$$\operatorname{osc}(D\varphi) \le 2 \sup_x |D\varphi(x)| \le 2R_{p,N} \operatorname{osc}(P^y_{p,N}\varphi) \le 2R_{p,N}\, \varrho(P^y_{p,N}) \operatorname{osc}(\varphi).$$

Applying the conditional bound in Eq. (28) and then taking expectations yields

$$\left\| \Phi_{p,N}(\eta^K_p)(\varphi) - \Phi_{p,N}(\tilde{\eta}^K_p)(\varphi) \right\|_{L_2} \le \frac{2\alpha_p R^2_{p,N}\, \varrho(P^y_{p,N})}{\sqrt{K}} \operatorname{osc}(\varphi).$$

Finally, the triangle inequality in L_2 sums these propagated errors,

$$\|\eta^K_N(\varphi) - \eta_N(\varphi)\|_{L_2} \le \frac{2 \operatorname{osc}(\varphi)}{\sqrt{K}} \sum_{p=0}^{N} \alpha_p R^2_{p,N}\, \varrho(P^y_{p,N}).$$

Since $\eta_N = \pi^{\star,(N)}$ by Step 1, this is Eq. (12) with the stated constant C_N. □

A.4 Proof of Remark 1

Under the assumptions of Remark 1, $X_1|X_t = x \sim \mathcal{N}(\mu^\star_{1|t}(x), \sigma^2_{1|t} I)$, so the exact twist is the Gaussian convolution

$$h_t(x) = \int \mathcal{N}(y; x_1, \sigma_y^2 I)\, \mathcal{N}(x_1; \mu^\star_{1|t}(x), \sigma^2_{1|t} I)\, \mathrm{d}x_1 = \mathcal{N}\big(y; \mu^\star_{1|t}(x), (\sigma_y^2 + \sigma^2_{1|t}) I\big).$$

The plug-in twist is $\hat{h}_t(x) = \mathcal{N}(y; \mu^\star_{1|t}(x), \sigma_y^2 I)$, so for any $\beta > 0$,

$$\hat{h}_t(x)^\beta = (2\pi\sigma_y^2)^{-\beta d/2} \exp\Big(- \frac{\beta \, \|y - \mu^\star_{1|t}(x)\|^2}{2\sigma_y^2} \Big).$$

With $\beta = \beta^\star(t) = \sigma_y^2/(\sigma_y^2 + \sigma^2_{1|t})$, the exponent equals $-\|y - \mu^\star_{1|t}(x)\|^2/\{2(\sigma_y^2 + \sigma^2_{1|t})\}$; hence $\hat{h}_t(x)^{\beta^\star(t)} = c_t\, h_t(x)$ with $c_t = (2\pi\sigma_y^2)^{-\beta^\star(t) d/2}\{2\pi(\sigma_y^2 + \sigma^2_{1|t})\}^{d/2}$ independent of x. Taking logarithms and gradients gives $\nabla \log \hat{h}_t^{\beta^\star(t)} = \nabla \log h_t$, so with $w_g = 1$ the guidance term in Eq. (10) equals the exact h-transform drift correction $g(t)^2 \nabla \log h_t$. The factors $c_{t_n}/c_{t_{n-1}}$ multiply every incremental weight (11) by the same particle-independent constant and cancel after normalization, so the tempered plug-in twist and the exact twist induce the same sampler.

The result also extends to observing only a subset of the coordinates of x_1. In the model $y = Ax_1 + \eta$ with $\eta \sim \mathcal{N}(0, \sigma_y^2 I)$, the matrix A selects the observed coordinates, such as the unmasked pixels of an image. Since $AA^\top = I$ in the observation space, the same computation gives $h_t(x) = \mathcal{N}\big(y; A\mu^\star_{1|t}(x), (\sigma_y^2 + \sigma^2_{1|t}) I\big)$, so $\hat{h}_t(x)^{\beta^\star(t)} \propto h_t(x)$ holds with the same $\beta^\star(t)$.

B Pretraining details

This appendix provides the SB pretraining convention assumed by the main text. We follow the simulation-free score-and-flow matching (SF2M) framework [18] but with a different parameterization, detailed below.

Score and flow matching. Under a Brownian reference with diffusion coefficient σ, the Schrödinger bridge's endpoint coupling is the entropic optimal transport (OT) coupling with squared Euclidean cost and regularization strength $2\sigma^2$. For endpoints (x_0, x_1) drawn from this coupling and $t \sim \mathrm{Unif}(0, 1)$, the corresponding Brownian bridge sample is

$$x_t = (1 - t)x_0 + t x_1 + \sigma\sqrt{t(1 - t)}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I).$$

Writing $m_t = (1-t)x_0 + tx_1$, its conditional probability-flow velocity and score are

$$u_t^\circ(x \mid x_0, x_1) = x_1 - x_0 + \frac{1-2t}{2t(1-t)}(x - m_t), \qquad s_t^\circ(x \mid x_0, x_1) = -\frac{x - m_t}{\sigma^2 t(1-t)}.$$

The standard SF2M objective jointly regresses these two targets [18],

$$\mathbb{E}\big[\|v_\theta(t, X_t) - u_t^\circ(X_t \mid X_0, X_1)\|^2 + w(t)^2 \|s_\theta(t, X_t) - s_t^\circ(X_t \mid X_0, X_1)\|^2\big],$$

where $w(t) > 0$ weights the score loss and the expectation is over the pair (X_0, X_1), time, and bridge noise. In our implementation, we approximate the endpoint coupling using minibatch exact OT, following Tong et al. [18].

Displacement and noise parameterization. Holding x_0, x_1, and ε fixed, differentiating the interpolation in t gives the conditional probability-flow velocity

$$u_t^\circ(x_t \mid x_0, x_1) = \underbrace{x_1 - x_0}_{\text{displacement } D} + \underbrace{\sigma \frac{1-2t}{2\sqrt{t(1-t)}}\varepsilon}_{\text{noise correction term}},$$

whose noise correction coefficient $\sigma(1-2t)/(2\sqrt{t(1-t)})$ diverges as $t \to 0$ or $t \to 1$. Directly regressing this velocity would inherit this divergence, so we predict D and ε separately and reconstruct the velocity $v_\theta(t, x)$ analytically. We instantiate a displacement channel $d_\theta(t, x)$ targeting $D = x_1 - x_0$ and a noise channel $\varepsilon_\theta(t, x)$ targeting ε. We consider a *joint* model with a shared UNet backbone and two heads, and a *dual* model that uses two independent UNets. Both are trained with the same loss, and experiments use the joint version for efficiency unless noted. The loss is

$$\mathbb{E}\left[\|d_\theta(t, x_t) - (x_1 - x_0)\|^2 + \lambda(t)\|\varepsilon_\theta(t, x_t) - \varepsilon\|^2\right],$$

where $\lambda(t) > 0$ weights the noise regression and the expectation is over pairs (x_0, x_1), times, and Gaussian noise.

Recovering velocity and score. The noise head is converted to the score convention used in the main text, $s_\theta(t, x) \approx \nabla \log p_t(x)$. Since $\nabla_{x_t} \log p(x_t|x_0, x_1) = -\varepsilon/(\sigma\sqrt{t(1-t)})$, the noise head converts to the score as

$$s_\theta(t, x) = -\frac{\varepsilon_\theta(t, x)}{\sigma\sqrt{t(1-t)}}.$$

The velocity is reconstructed from the displacement identity

$$\mathbb{E}[D \mid X_t = x] = v_t^\star(x) + \frac{1-2t}{2}\sigma^2 s_t^\star(x).$$

Thus the implemented velocity channel is

$$v_\theta(t, x) = d_\theta(t, x) - \frac{1-2t}{2}\sigma^2 s_\theta(t, x).$$

With this convention, the endpoint predictor used by BRIDGETWIST is exactly Eq. (8). If an implementation stores the scaled score $s_{\mathrm{sc}}(t, x) = (\sigma^2/2)s_\theta(t, x)$, then Eq. (8) is evaluated as

$$\hat{\mu}_{1|t}(x) = x + (1-t)v_\theta(t, x) + (1-t)s_{\mathrm{sc}}(t, x).$$

C Implementation details

C.1 Two-dimensional toy benchmark

For the toy setup in Section 4.1, we use the independent coupling $X_0 \sim q_0$, $X_1 \sim q_1$ and the Brownian-bridge interpolant $X_t = (1-t)X_0 + tX_1 + \sigma\sqrt{t(1-t)}\,\epsilon$. Under this interpolant the component marginal is $p_t(x|k) = \mathcal{N}(x; t\mu_k, \sigma_t^2 I)$ with $\sigma_t^2 = (1-t)^2 + \sigma^2 t(1-t) + t^2/4$, and the responsibilities are

$$\gamma_k(t, x) = \frac{\pi_k \mathcal{N}(x; t\mu_k, \sigma_t^2 I)}{\sum_j \pi_j \mathcal{N}(x; t\mu_j, \sigma_t^2 I)}.$$

The analytic score and probability-flow velocity used by BRIDGETWIST, DPS-SB, and IS are

$$s_t^\star(x) = \sum_k \gamma_k(t,x)\,\frac{-(x - t\mu_k)}{\sigma_t^2}, \qquad v_t^\star(x) = \sum_k \gamma_k(t,x)\left[\mu_k + \frac{\dot\sigma_t^2}{2\sigma_t^2}(x - t\mu_k)\right],$$

with $\dot\sigma_t^2 = -2(1-t) + \sigma^2(1-2t) + t/2$.

FMPS and TFTF instead use the deterministic flow-matching path between the same distributions q_0 and q_1, whose component variance is $\tilde\sigma_t^2 = (1-t)^2 + t^2/4$ (with $\tilde\gamma_k$ defined by substituting $\tilde\sigma_t^2$ above) and whose velocity is

$$v_t^{\mathrm{FM}}(x) = \sum_k \tilde\gamma_k(t,x)\left[\mu_k + \frac{t/4 - (1-t)}{\tilde\sigma_t^2}(x - t\mu_k)\right].$$

For TDS, we use the standard variance-preserving (VP) diffusion marginal with the same endpoint mixture q_1:

$$X_t = a_t X_1 + r_t Z, \qquad a_t = \sin(\pi t/2), \quad r_t = \cos(\pi t/2), \quad Z \sim \mathcal{N}(0, I).$$

Thus, $p_t(x|k) = \mathcal{N}(x; a_t\mu_k, V_t I)$ with $V_t = a_t^2/4 + r_t^2$, and

$$\gamma_k^{\mathrm{VP}}(t,x) = \frac{\pi_k\,\mathcal{N}(x; a_t\mu_k, V_t I)}{\sum_j \pi_j\,\mathcal{N}(x; a_t\mu_j, V_t I)}.$$

The analytic VP score used by TDS is

$$s_t^{\mathrm{VP}}(x) = \sum_k \gamma_k^{\mathrm{VP}}(t,x)\frac{-(x - a_t\mu_k)}{V_t}.$$

where $\dot a_t = (\pi/2) r_t$, $\dot r_t = -(\pi/2) a_t$, and $\dot V_t = 2(a_t\dot a_t/4 + r_t\dot r_t)$. The Tweedie endpoint predictor in TDS is

$$\mu_{1|t}^{\mathrm{VP}}(x) = \sum_k \gamma_k^{\mathrm{VP}}(t,x)\left[\mu_k + \frac{a_t/4}{V_t}(x - a_t\mu_k)\right].$$

C.2 Unconditional models and likelihoods

The unconditional SB models used by BRIDGETWIST, DPS-SB, and IS are pretrained as described in Appendix B, with target q_1 given by MNIST, CIFAR-10, or CelebA-HQ, source $q_0 = \mathcal{N}(0, I)$ in pixel space, and Brownian reference $\sigma = 1$ for MNIST and CIFAR-10 and $\sigma = 0.25$ for CelebA-HQ. For a fair comparison, FMPS [6] and TFTF [5] use the same unconditional flow matching checkpoints as TFTF, rather than the velocity field learned by the SB. We initialize the displacement head $d_\theta(t,x)$ in our SB pretraining from these checkpoints, so the compared methods have approximately matched network capacity.

Base models. For MNIST, the initialization is the pretrained U-Net used by TFTF [5]. For CIFAR-10 and CelebA-HQ, the initialization is the pretrained rectified flow model of Liu et al. [50] from its public RectifiedFlow release.[2]

Table 9 reports the parameter counts and the unconditional FID (10k samples against the training set) of the priors used by the compared methods. The SB prior adds one score head to the rectified-flow architecture, so the parameter counts match within 0.1%, and its unconditional FID is slightly worse than that of the flow and diffusion checkpoints; the conditional gains of BRIDGETWIST therefore do not come from a larger or better base model. The SB prior is initialized from the TFTF checkpoint and fine-tuned for $40{,}000$ steps with the additional score head, which takes about 5.5 GPU hours on an RTX 5090 for CIFAR-10, a one-time overhead relative to using a pretrained flow or diffusion model directly.

[2] https://github.com/gnobitab/RectifiedFlow

Table 9: Parameter count and unconditional FID (10k samples) of the pretrained priors. NA marks the absence of a released checkpoint.

Prior model	MNIST params / FID	CIFAR-10 params / FID	CelebA-HQ params / FID
Rectified flow (TFTF checkpoint)	14.9777M / 5.22	61.8044M / 7.26	65.5745M / 48.40
DDPM / iDDPM (TDS checkpoint)	14.4154M / 2.27	52.5464M / 7.33	NA
SB (ours)	14.9782M / 6.53	61.8079M / 8.49	65.6160M / 51.54

Likelihood models. For MNIST and CIFAR-10 class-conditional sampling, the likelihood classifier is a frozen ResNet-50 [51] from the minimal-diffusion release.[3] For inpainting, super-resolution, and deblurring, we use the Gaussian likelihood $L_y(x_1) = \mathcal{N}(y; Ax_1, \sigma_y^2 I)$. The observation operator A selects the observed pixels for inpainting, performs $4\times$ bicubic downsampling for super-resolution, and applies a 9×9 Gaussian blur with standard deviation 2 for deblurring. The observation noise standard deviation is $\sigma_y = 0.05$ for MNIST inpainting, 0.1 for CIFAR-10 inpainting, and 0.03 for both CelebA-HQ restoration tasks. For CelebA-HQ text-prompted image generation, the likelihood is computed with a ConvNext-XXLarge CLIP model from OpenCLIP [45, 52, 53].[4] The likelihood takes the form $L_y(x_1) \propto \exp\{\lambda\, \mathrm{sim}(\phi_I(x_1), \phi_T(y))\}$, where ϕ_I and ϕ_T are image and text embeddings, sim is cosine similarity, and λ sets the energy scale.

External evaluators. For MNIST, external accuracy is measured by a separately trained Super Learner classifier.[5] For CIFAR-10, external accuracy is measured by a DenseNet-121 classifier [54] from a public CIFAR-10 model release.[6] For CelebA-HQ, external prompt alignment is measured by the attribute classifiers released with TransEditor [55].[7]

C.3 Hyperparameters and metrics

Toy example. BRIDGETWIST uses guidance strength $w_g = 0.5$, bridge-shaped diffusion schedule $g(t) = c_g \sigma \sqrt{t(1-t) + \delta_g}$ with $c_g = 1$, ESS threshold $\rho = 0.8$, and systematic resampling activated from $t = 0.1$ onward, with the small constants $\delta_g = \varepsilon = 10^{-6}$. Guidance-baseline strengths are tuned by grid search: $r = 0.2$ for FMPS and $\zeta = 0.5$ for DPS-SB. TFTF uses the default three-phase schedule [5].

Image experiments. BRIDGETWIST uses guidance strength $w_g = 1$ for class-conditional sampling and text-prompted image generation and $w_g = 2$ for inpainting, the same bridge-shaped schedule with $c_g = 4$, and ESS threshold $\rho = 0.5$; the remaining settings are same to the toy example. Guidance-baseline strengths are still tuned by grid search: $r = 1$ for FMPS and $\zeta = 200$ for DPS-SB on MNIST and CIFAR-10; $r = 0.003$ for FMPS and $\zeta = 50$ for DPS-SB on CelebA-HQ.

CIFAR-10 inpainting baselines. DAPS and SITCOM use the unconditional iDDPM prior used by TDS; PnP-Flow uses the rectified-flow prior used by TFTF. DAPS uses 200 annealing steps, five ODE substeps and 100 Langevin updates per annealing step, a Langevin learning rate of 5×10^{-5}, and $\tau = 0.01$. SITCOM uses 20 DDIM steps with up to 30 Adam iterations per step, a learning rate of 0.02, a projection scale of 0.5, and a noise-level parameter of 0.1. PnP-Flow uses 100 steps and a learning rate of 1.0, with $\alpha = 0.5$ for half-image masks and $\alpha = 0.01$ for random 50% masks.

CelebA-HQ restoration baselines. PnP-Flow, Flower, FlowDPS, TFTF, and FMPS share the rectified-flow prior in Table 9, their native model class. DAPS and SITCOM are excluded from this comparison because no compatible released CelebA-HQ diffusion checkpoint is available. PnP-Flow uses 100 steps and a learning rate of 1.0, with $\alpha = 0.01$ for deblurring and $\alpha = 0.3$ for super-resolution. Flower uses 100 steps with up to 50 conjugate-gradient iterations per step. The stochastic time progression is $x \leftarrow (t + \delta)x^\star + (1 - t - \delta)z_0$, with $\delta = 1/100$ and time grid $t = \delta i$ for $i = 0, \ldots, 99$. We retain the original time parameterization and the $[-1, 1]$ image convention without additional output rescaling. FlowDPS uses 28 solver iterations, each with three data-consistency

[3] https://github.com/VSehwag/minimal-diffusion
[4] https://github.com/mlfoundations/open_clip
[5] https://github.com/Curt-Park/handwritten_digit_recognition
[6] https://github.com/huyvnphan/PyTorch_CIFAR10
[7] https://github.com/BillyXYB/TransEditor

gradient updates of step size 15. The baselines use their released configurations, with no additional tuning of learning rates or other hyperparameters on these two restoration tasks.

Metrics. All metrics are computed on endpoint samples after resampling under the final particle weights. For class-conditional sampling, each metric is averaged over K particles, 10 classes, and 100 replications per-class. Density and Coverage [44] are computed per class between the generated samples and 1,000 real training images of that class in Inception feature space with $k = 5$ nearest neighbours, and averaged over classes. For text-prompted image generation, each metric is averaged over K particles, 40 prompts, and 5 replications per-prompt. LPIPS diversity is the mean pairwise LPIPS distance [46] among the samples generated for the same prompt. For inpainting, we use half-image and random 50% masks and evaluate SSIM on the full reconstruction. For the CelebA-HQ inverse problems, PSNR and SSIM are averaged over 100 test images and three independent runs per image, with standard deviations computed over all images and runs.

The main image experiments were conducted on an NVIDIA RTX 5090 GPU.

D Additional results

D.1 Computation cost

Table 10 reports the number of forward and backward backbone evaluations (NFEs), the wall-clock time, and the peak GPU memory of one CIFAR-10 class-conditional run at $K = 64$ and $N = 800$, using the configurations of Table 2. BRIDGETWIST obtains (v_θ, s_θ) from one shared-backbone evaluation and performs one input-gradient backward pass per particle and time step; its wall-clock time and peak memory are comparable to those of the gradient-based baselines, and its $\mathcal{O}(KN)$ cost is of the same order as that of TFTF, TDS, and FMPS.

Table 10: Computation cost of one CIFAR-10 class-conditional run at $K = 64$ and $N = 800$: forward and backward backbone evaluations, wall-clock time, and peak GPU memory.

Method	forward NFE	backward NFE	time (s)	memory (GB)
FMPS	51,200	51,200	48.1	6.9
TFTF	51,200	22,336	46.1	6.9
DPS-SB	51,200	51,200	50.9	6.7
IS	51,200	0	20.7	1.3
TDS	51,200	51,200	45.3	7.0
BRIDGETWIST	51,264	51,264	50.1	6.7

D.2 Joint particle and time-step allocation

We vary the particle count K and time-step count N jointly for BRIDGETWIST under total forward-plus-backward backbone evaluation (NFE) budgets $B \in \{25632, 102528\}$. For each $N \in \{100, 200, 400, 800\}$, we choose $K = \lfloor B/[2(N+1)] \rfloor$, the largest integer particle count with $\mathrm{NFE} = 2(N+1)K \le B$; the actual counts are reported in Table 11.

Figure 5 and Table 11 report the quality metrics and computational costs, respectively. We evaluate the K endpoint samples after systematic resampling from the terminal weights. The results show that the allocation between K and N affects quality under a fixed computational budget, and a relatively large time-step count N (e.g., $N = 400$) can be beneficial.

D.3 Qualitative samples

Figure 6 compares MNIST half-mask inpainting, and Figure 7 shows BRIDGETWIST reconstructions for the super-resolution and deblurring tasks of Table 5.

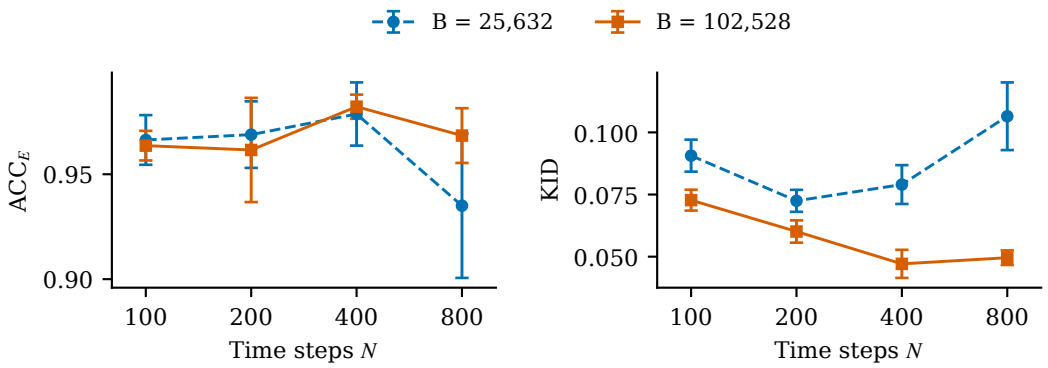


Figure 5: CIFAR-10 quality under two total backbone-NFE ceilings B. Points and error bars show the mean and sample standard deviation based on ten classes and five repetitions per-class. Higher ACC_E and lower KID are preferred.

Table 11: Particle allocation and sampling costs. Memory gives mean peak allocation.

N	K	Actual NFE	Time (s)	Memory (GiB)
		$B = 25{,}632$		
100	126	25,452	$45.81_{\pm 0.07}$	6.708
200	63	25,326	$44.49_{\pm 0.19}$	6.678
400	31	24,862	$42.16_{\pm 0.07}$	3.468
800	16	25,632	$80.11_{\pm 0.02}$	2.175
		$B = 102{,}528$		
100	507	102,414	$177.80_{\pm 0.20}$	6.782
200	255	102,510	$189.03_{\pm 20.72}$	6.732
400	127	101,854	$183.21_{\pm 0.18}$	6.708
800	64	102,528	$178.78_{\pm 0.28}$	6.701

D.4 Expanded CIFAR-10 inpainting comparison

Table 12 adds DAPS, SITCOM, and TDS to the CIFAR-10 inpainting comparison. The methods use the prior families and settings in Appendix C.3. BRIDGETWIST attains the highest SSIM for both masks.

Table 12: CIFAR-10 inpainting quality and cost. Time is per image.

Method	Time (s)	Peak memory (GB)	SSIM, half ↑	SSIM, random ↑
DAPS	20.21	0.57	0.626	0.908
SITCOM	3.82	0.71	0.645	0.907
TDS	2.26	1.78	0.630	0.927
PnP-Flow	5.08	0.60	0.623	0.940
FMPS	2.27	1.64	0.240	0.554
TFTF	2.56	1.86	0.532	0.875
DPS-SB	2.44	1.59	0.518	0.782
BRIDGETWIST	2.42	1.59	**0.655**	**0.943**

D.5 Calibration of the endpoint predictor

We measure the gap between the plug-in predictor $\hat{\mu}_{1|t}$ and $\mathbb{E}[X_1|X_t]$ along the trajectory. For a state x_t, we estimate $\mathbb{E}[X_1|X_t = x_t]$ by the Monte Carlo average $\bar{\mu}_{\mathrm{MC}}$ of 128 suffix trajectories of the prior dynamics started at (t, x_t) and report $\mathrm{err}^2(\hat{\mu}_{1|t}) = \|\hat{\mu}_{1|t}(x_t) - \bar{\mu}_{\mathrm{MC}}\|^2/d$. Table 13 shows that the error is largest near $t = 0$ and decays with t. Writing $b^\star$ for the SB drift and $b_{\theta,g}$ for the sampler

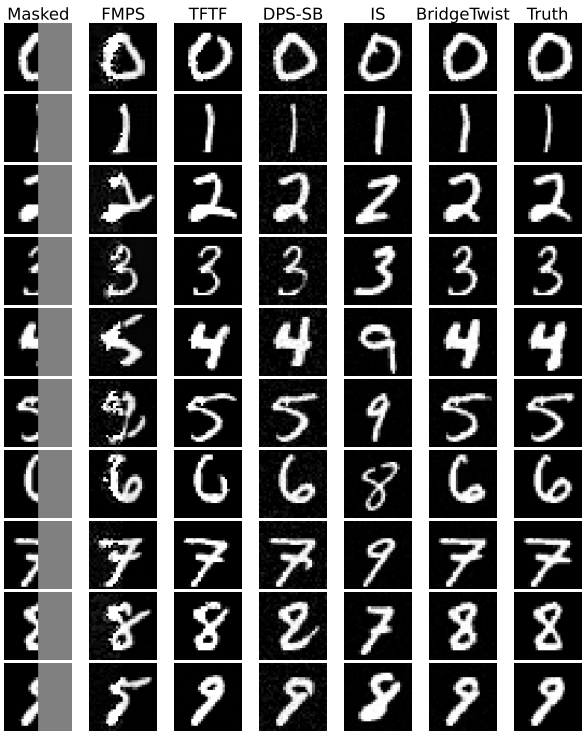


Figure 6: MNIST half-mask inpainting comparison. Each row shows one digit example; columns show the masked input, FMPS, TFTF, DPS-SB, IS, BRIDGETWIST, and the ground truth.

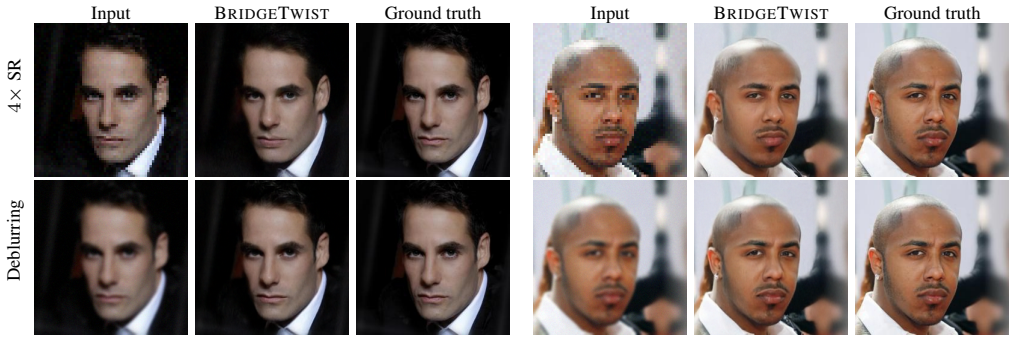


Figure 7: CelebA-HQ reconstructions from BRIDGETWIST for $4\times$ super-resolution (top) and Gaussian deblurring (bottom). Each triplet shows the observation, the BRIDGETWIST reconstruction, and the ground truth.

drift, the gap has the form

$$\hat{\mu}_{1|t}(x) - \mathbb{E}[X_1 \mid X_t = x] = \mathbb{E}\Big[\int_t^1 \big(b_t^\star(x) - b_{\theta,g}(u, X_u)\big)\, \mathrm{d}u \;\Big|\; X_t = x \Big],$$

so as $t \to 1$, the interval $(t, 1)$ shrinks and X_u stays near x within it, and the integrand and the range of integration vanish together.

Table 13: Calibration error $\mathrm{err}^2(\hat{\mu}_{1|t})$ of the endpoint predictor against Monte Carlo estimates of $\mathbb{E}[X_1|X_t]$ at different times t.

t	0.05	0.1	0.3	0.5	0.7	0.9	0.95
Toy example	0.5572	0.4587	0.1303	0.0193	0.0034	0.00066	0.00031
CIFAR-10	0.00334	0.00164	0.00129	0.00157	0.00072	0.00044	0.00052

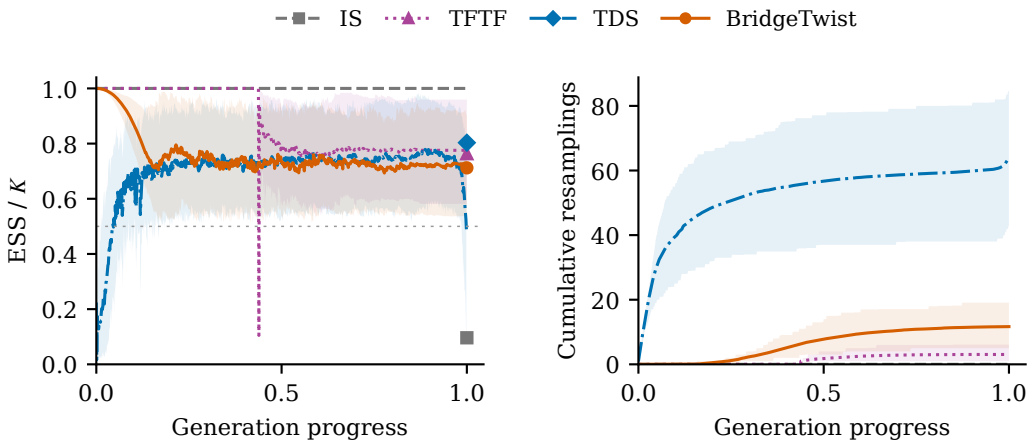


Figure 8: CIFAR-10 ESS and resampling trajectories at $K = 64$ and $N = 800$. Curves are means over ten classes and ten repetitions per class; shading gives the 10th–90th percentile range across the 100 runs. Left: pre-resampling ESS/K, with the 0.5 threshold and separate markers for terminal ESS after the final weight update. Right: cumulative counts of resampling events.

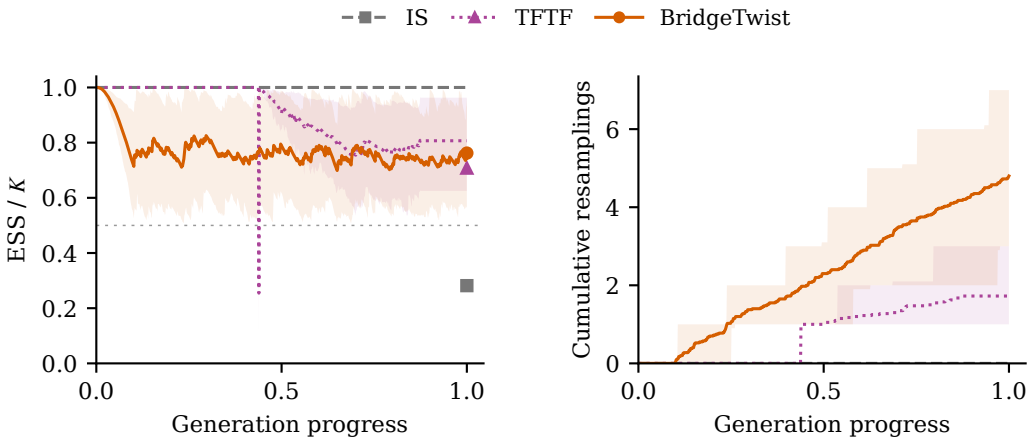


Figure 9: CelebA-HQ ESS and resampling trajectories at $K = 25$ and $N = 800$. Curves are means over 40 attribute prompts with one run per prompt, and shading gives the 10th–90th percentile range across these prompt trials. Left: pre-resampling ESS/K, with the 0.5 threshold and separate terminal ESS markers. Right: cumulative counts of resampling events.

D.6 SMC diagnostics

We record the ESS and resampling trajectories for CIFAR-10 class-conditional sampling and CelebA-HQ text-prompted image generation in Figures 8 and 9, respectively. The ESS threshold is 0.5; BRIDGETWIST enables resampling after the first 10% of the path, while TFTF resamples only in its stochastic phase. IS has uniform intermediate weights and weights samples only at the endpoint, so its intermediate ESS/K is one by construction.

Table 14 summarizes for both datasets. On CIFAR-10, BRIDGETWIST maintains a comparable time-averaged ESS to TDS with substantially fewer resampling events. On CelebA-HQ, BRIDGETWIST has a higher terminal ESS than IS and TFTF, while TFTF has a higher time-averaged ESS and fewer resampling events than BRIDGETWIST.

Table 14: SMC diagnostics along the full trajectories. A dash marks the absence of a compatible checkpoint.

	CIFAR-10			CelebA-HQ		
Method	step-avg. ESS/K	resamplings	final ESS/K	step-avg. ESS/K	resamplings	final ESS/K
IS	1.000	0	0.096	1.000	0	0.282
TFTF	0.875	3.05	0.763	0.902	1.73	0.708
TDS	0.705	64.25	0.803	–	–	–
BRIDGETWIST	0.756	11.67	0.712	0.770	4.80	0.762

D.7 Tempering in TDS and TFTF

We apply our linear tempering schedule $\beta(t) = t$ to the twists in TDS and TFTF, retaining each method's prior and guided-proposal construction. Table 15 reports the comparisons on CIFAR-10. All runs use $K = 64$ particles and $N = 800$ steps. Tempering improves TDS, consistent with correcting its plug-in over-confidence, and it slightly reduces TFTF's accuracy, consistent with TFTF's early unconditional phase already suppressing guidance. TDS with tempering remains below BRIDGETWIST, which additionally uses an SB-specific bridge-shaped inference schedule, a flexibility with no counterpart under TDS.

Table 15: TDS, TFTF, their linearly tempered variants, and BRIDGETWIST on CIFAR-10.

Method	$\text{ACC}_L \uparrow$	$\text{ACC}_E \uparrow$	KID $\downarrow$
TDS	0.947	0.869	0.129
TDS + tempering	0.995	0.951	0.068
TFTF	0.986	0.934	0.057
TFTF + tempering	0.984	0.923	0.051
BRIDGETWIST	**0.997**	**0.979**	**0.040**

D.8 Second-order twist extension

The first-order twist evaluates the likelihood at the conditional endpoint mean estimate and discards conditional endpoint uncertainty. A second-order approximation represents the conditional endpoint law by a Gaussian with mean $\hat{\mu}_{1|t}(x)$ and covariance $\widehat{\Sigma}_{1|t}(x)$. For a linear Gaussian observation $y = Ax_1 + \eta$, where $y \in \mathbb{R}^m$, $x_1 \in \mathbb{R}^d$, $A \in \mathbb{R}^{m\times d}$, and $\eta \sim \mathcal{N}(0, R)$ with $R \succ 0$, integrating the likelihood gives

$$\hat{h}_t^{(2)}(x) = \int \mathcal{N}(y; Az, R)\, \mathcal{N}(z; \hat{\mu}_{1|t}(x), \widehat{\Sigma}_{1|t}(x))\, \mathrm{d}z = \mathcal{N}(y; A\hat{\mu}_{1|t}(x), R + A\widehat{\Sigma}_{1|t}(x)A^\top).$$

Thus, endpoint uncertainty increases the covariance in the predictive likelihood.

For the two-dimensional toy in Appendix C.1, $A = I$ and $R = \sigma_y^2 I$. We compare the first-order twist with second-order twists using either the exact conditional covariance or a Jacobian-based covariance estimate. For the estimated variant, let $\mu^\star_{M|t}(x) = x + \tau_t^2 s_t^\star(x)$ denote $\mathbb{E}[M_t|X_t = x]$, where $\tau_t^2 = \sigma^2 t(1-t)$. We use

$$\begin{aligned}
\Sigma_{M|t}(x) &:= \mathrm{Cov}(M_t \mid X_t = x) = \tau_t^2\big[I + \tau_t^2 \nabla_x s_t^\star(x)\big], \\
G_t(x) &:= I + (1-t)\left[\nabla_z v_t^\star(z) + \frac{\sigma^2}{2}(1-2t)\nabla_z s_t^\star(z)\right]_{z=\mu^\star_{M|t}(x)}, \\
\widehat{\Sigma}_{1|t}(x) &:= G_t(x)\Sigma_{M|t}(x)G_t(x)^\top.
\end{aligned}$$

All Jacobians are evaluated analytically in this two-dimensional experiment.

Table 16 reports the comparison at $K = 2000$ and $N = 100$, with 50 independent repetitions per observation-noise level. Both second-order variants improve the results at the tighter observation-noise levels, while the differences are small at $\sigma_y \in \{0.5, 1.0\}$. Constructing the full covariance in high-dimensional image spaces requires additional Jacobian operations, so the image experiments retain the first-order twist.

Table 16: First- and second-order twists on the two-dimensional toy. Entries are the square root of the mean MMD^2, with smaller values preferred.

Twist / σ_y	1.0	0.5	0.3	0.1	0.05
First-order	0.032	0.033	0.134	0.062	0.059
Second-order, exact covariance	0.032	0.031	0.037	0.025	0.016
Second-order, estimated covariance	0.033	0.031	0.037	0.028	0.020

D.9 Conditional transition with a non-Gaussian source

To illustrate the applicability of BRIDGETWIST to non-Gaussian sources, we evaluate BRIDGETWIST on the PHATE embryoid-body scRNA-seq data [56] (Mendeley v6n743h5ng). We transport an early capture block (T4_5C) to a late block (T6_7D) while conditioning on a high proliferation-marker score at the endpoint, using 2048 cells per block, 30 principal components of 2000 highly variable genes, $K = 2048$, and $N = 150$. We use the posterior proxy inferred from the official Waddington-OT transport map [57] as the reference, which is independent of all samplers.

To target cells with high proliferation-marker expression, we construct a single score from the expression of MKI67, TOP2A, PCNA, CENPF, TYMS, and UBE2C. We condition on a noisy observation of this score at its 95th percentile under the Waddington-OT descendant distribution.

Table 17 reports the energy distance and the sliced Wasserstein-2 distance to the reference; BRIDGETWIST attains the lowest values of both, and TFTF degrades severely. This stochastic transport from a non-Gaussian source illustrates an SB-specific use case for BRIDGETWIST.

Table 17: Conditional transition on PHATE scRNA-seq data with a non-Gaussian source. Bold marks the best value in each column.

Method	Energy distance ↓	Sliced W_2 ↓
FMPS	0.0650	0.0599
TFTF	3.1958	0.4919
DPS-SB	0.0463	0.0550
IS	0.0701	0.0610
BRIDGETWIST	**0.0374**	**0.0470**

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes].
   Justification: The claims are stated in the abstract and Section 1; the theoretical scope is specified in Section 3.3, and limitations are discussed in Section 5.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes].
   Justification: Section 5 discusses the restriction to a Brownian reference, limitations of the plug-in twist, errors from prior learning and discretization, and the additional pretraining cost.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes].
   Justification: Appendix A gives the assumptions and proofs for Propositions 1 and 2, Theorem 1, and Remark 1.
   Guidelines:
   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes].
   Justification: Section 4 specifies tasks, baselines, particle counts, sampler steps, and metrics; Appendix C additionally provides analytic fields for the toy benchmark, the unconditional pretraining setup, conditional likelihoods, hyperparameters and settings, and compute resources.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes].
   Justification: A code release with pretrained checkpoints and scripts to reproduce the main experimental results is provided in the supplementary material.
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [Yes].
   Justification: Section 4 specifies the tasks, baselines, particle counts, sampler steps, and evaluation metrics. Appendix B describes the pretraining parameterization; Appendix C gives the pretrained models, likelihoods, sampler hyperparameters, baseline settings, and metric averaging conventions.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes].
   Justification: Section 4 reports the standard deviations of metrics based on multiple independent replications.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes].
   Justification: Appendix C states that all experiments were conducted on one NVIDIA RTX 5090 GPU.
   Guidelines:

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

9. **Code of ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?
   Answer: [Yes].
   Justification: The work uses existing public image datasets and the PHATE embryoid-body single-cell RNA-sequencing dataset for computational experiments, without recruiting participants or collecting new human-subject data.
   Guidelines:
   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**
    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
    Answer: [N/A].
    Justification: The paper presents a training-free sampler for pretrained Schrödinger bridge generative models. It inherits the broader-impact profile of generic conditional generative modeling without introducing new specific risks.
    Guidelines:
    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**
    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?
    Answer: [N/A].
    Justification: The paper does not release a high-risk foundation model or a scraped dataset.
    Guidelines:
    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes].
    Justification: All existing methods, datasets, and pretrained checkpoints used in this work are properly cited in Sections 1–4 and Appendix C; their licenses are documented in the README of the supplementary code release.
    Guidelines:
    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [Yes].
    Justification: The supplementary release contains the reproduction code and the pretrained SB checkpoints used to produce the reported results, documented in the README. No new datasets or benchmark assets are introduced.
    Guidelines:
    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [N/A].
    Justification: The experiments do not involve crowdsourcing or human subject studies.
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
    Answer: [N/A].
    Justification: The experiments do not involve crowdsourcing or human subject studies.
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**
    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
    Answer: [N/A].
    Justification: The core methodology does not use LLMs as an original or non-standard component.
    Guidelines:
    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.