LatentFlow: A General Framework for Conditioning Stochastic Processes

Anonymous authors
Paper under double-blind review

Abstract

Stochastic-process models are, as a rule, far easier to simulate than to condition. Non-linear observations, non-Gaussian likelihoods, black-box information, and global constraints all induce intractable conditional laws, requiring bespoke, model-specific constructions. We introduce LatentFlow, a single framework for conditioning stochastic processes, with no learned neural approximations and no training. Our starting point is to write the stochastic process as the deterministic image of a tractable latent innovation, $f_0 = T_\vartheta(\xi_0)$, with ξ_0 sampled from a simple reference distribution. This reduces process-level conditioning to latent-space inference: pull the likelihood back through T_ϑ, sample the resulting latent law with a tractable guided probability flow, and push the samples forward. This construction is provably exact at the level of the target law; in practice, approximation enters only through finite terminal noising, Monte Carlo guidance, and time discretisation of the continuous-time dynamics, each of which is explicit and systematically reducible. As LatentFlow is training-free, conditioning reduces to solving a single reverse-time SDE. This enables conditional sampling in seconds on a single desktop CPU across model classes that have never shared a scalable method: classical spatial priors, nonlinear stochastic dynamics, mechanistic models from the physical and life sciences, stochastic PDEs, heavy-tails and extremes, point and discrete-state processes, and neural or simulator-defined processes.

1 Introduction

Sampling from a stochastic process only requires running it forward; conditioning it, until now, has in many cases remained the preserve of bespoke, model-specific constructions. Examples include diffusion paths conditioned on fixed endpoints or partial observations (Delyon & Hu, 2006; Schauer et al., 2017); functions with shape (Riihimäki & Vehtari, 2010) or physics-informed (Chen et al., 2021; Hamelijnck et al., 2024) constraints; and simulator outputs conditioned on indirect measurements, rare events, or user-defined criteria (Botev & L'Ecuyer, 2020; Finzi et al., 2023). In each case the conditioning information cannot be absorbed into the model in closed form, demanding its own expensive or approximate sampler; for instance, those based on MCMC (Andrieu et al., 2003), SMC (Doucet et al., 2001), variational inference (Blei et al., 2017) or Laplace approximations (Rue et al., 2009).

Recently, Moss et al. (2026) introduced FlowGP, which addresses conditioning for Gaussian process (GP) priors under intractable information by recasting conditional sampling as a differential equation with closed-form Gaussian dynamics and a likelihood-dependent guidance term. The resulting sampler applies to any likelihood that can be evaluated pointwise, and requires no trained network to approximate the guidance field. Underlying the model is a simple mechanism: the GP sample is the linear, deterministic image of a Gaussian innovation, and conditioning is a re-weighting of that innovation's density by the pulled-back likelihood.

Our key observation is that Gaussianity of the stochastic process prior is not the essential ingredient in this construction; instead, it is the availability of a tractable *latent innovation representation* on which guided dynamics can be defined. Rather than constructing condition-

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

ing dynamics directly in the original process space, this allows us to express the process as the deterministic image of a tractable innovation variable, pull the conditioning information back through this generator, and sample the resulting reweighted law in latent space. The conditioned process is then obtained by pushing the guided samples in the latent space forward through the same generator. We refer to this method as LatentFlow. Crucially, our construction does not require learning a process prior, score model, or conditional generator (e.g., Song et al., 2021; Chung et al., 2023; Zammit-Mangion et al., 2025); the sampler is built directly from the known map T_ϑ and pointwise evaluations of the pulled-back likelihood. Our approach also extends directly to fully Bayesian inference for unknown model parameters: their prior-generating random variables can be included as additional latent innovations, yielding joint posterior samples of the process and its parameters; see Sec. 3.

We obtain formal theoretical guarantees for LatentFlow, and demonstrate its application to a wide variety of stochastic process priors. These include GPs; heavy-tailed Student-t and Cauchy-convolution processes; discrete-valued Potts processes; linear and nonlinear diffusion processes including a model for cell-differentiation, a FitzHugh–Nagumo model for excitable neurons, and a Heston stochastic volatility model; mechanistic models of SIR epidemics and Lotka–Volterra predator–prey dynamics; and Allen–Cahn and advection–diffusion SPDEs. See Fig. 1 for several examples. Until now, beyond simple or contrived cases, conditioning each of these processes required either low-fidelity approximations or expensive computation. We sample in single-seconds time on a single desktop CPU, providing the first general method for conditioning stochastic processes in real time which is exact up to an explicit and systematically reducible numerical error.

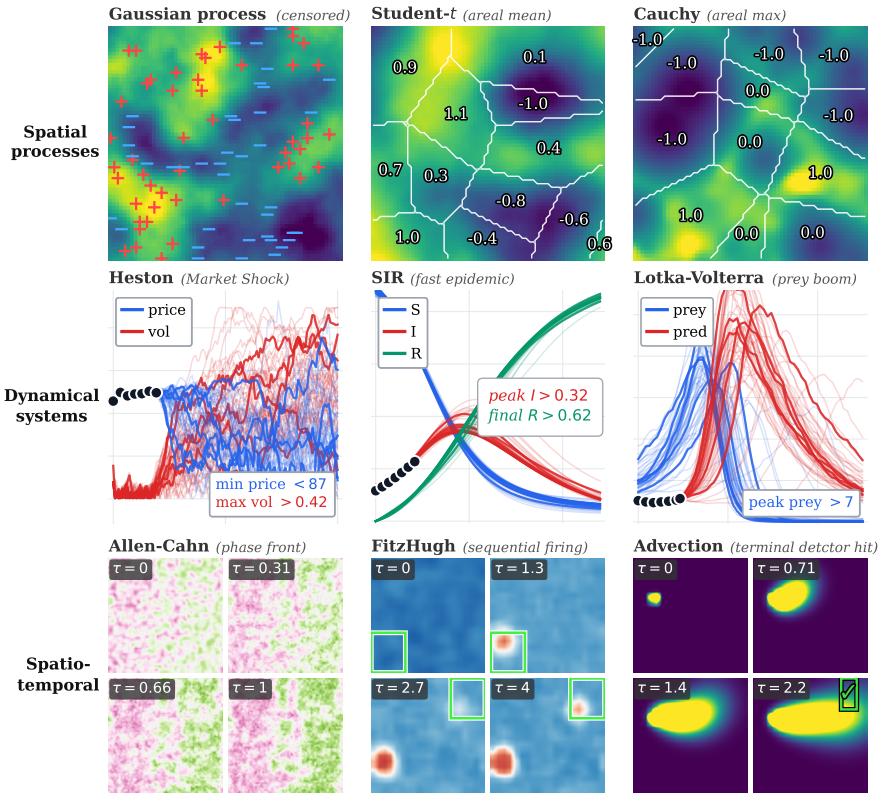


Figure 1: LatentFlow across diverse stochastic-process priors and conditions. **Top**: spatial processes conditioned on censored point measurements for a Gaussian process; prescribed areal means for a Student-t process; and prescribed areal maxima for a Cauchy convolution field. **Centre**: SDEs conditioned on observations and events. A Heston model under a joint market-crash and volatility spike; an SIR model under a fast epidemic peak; and a Lotka–Volterra system given a large prey boom. **Bottom**: Spatio-temporal PDEs. Allen–Cahn conditioned on a prescribed phase front (negative left, positive right) at final time; FitzHugh–Nagumo conditioned on sequential source firing (bottom-left then top-right); and an advection–diffusion plume conditioned on reaching a downstream detector (box).

2 Conditioning via a Latent Representation

Let $f \in \mathcal{E}$ denote a stochastic process, viewed as a random element in a measurable space $\mathcal{E}$, and let $f_0 \in E_m$ denote a finite-dimensional discretisation of f. In typical examples, $E_m \subseteq \mathbb{R}^m$ or $E_m \subseteq \mathbb{R}^{md}$, where m is the number of input locations and d is the state dimension at each location. Let $p_\vartheta(f_0)$ denote the prior density of f_0 on E_m, indexed by model parameters ϑ. We are interested in sampling the conditional distribution of this process, given some additional information $\mathcal{C}$ encoded by a likelihood $L_\mathcal{C} : E_m \to [0, \infty)$. The desired conditional density is thus given by

$$\pi_\vartheta(f_0 \mid \mathcal{C}) = \frac{L_\mathcal{C}(f_0)\, p_\vartheta(f_0)}{Z_\vartheta}, \qquad Z_\vartheta = \int_{E_m} L_\mathcal{C}(f_0)\, p_\vartheta(f_0)\, \mathrm{d}f_0. \tag{1}$$

Our key assumption is that p_ϑ can be obtained by transforming a simple reference density; see Remark B.1 for a measure-theoretic generalisation, which also applies when the prior is singular or discrete.

Assumption 2.1. There exists a latent dimension m_ξ, a reference density r on $\mathbb{R}^{m_\xi}$, and a map $T_\vartheta : \mathbb{R}^{m_\xi} \to E_m$ such that, if $\xi_0 \sim r$, then $f_0 = T_\vartheta(\xi_0)$ has density p_ϑ.

Assumption 2.1 is very mild; after discretisation, it holds for essentially any process of interest. Consider a scalar target with continuous distribution function F. By the probability integral transform, the variable $F^{-1}(\mathrm{R}(\xi_0))$ with $\xi_0 \sim r$ has distribution F, where R is the distribution function of r. The Rosenblatt transform (Rosenblatt, 1952) generalises this to multivariate f_0 and writes *any* absolutely continuous law on E_m as the deterministic image of some $\xi_0 \sim r$. The choice of r is arbitrary; we fix $r(\xi_0) = \mathcal{N}(\xi_0; 0, I_{m_\xi})$ throughout.

Once such a representation is available, conditioning the process on $\mathcal{C}$ can be transferred to the latent space via the *latent likelihood* $\Lambda_{\vartheta,\mathcal{C}}(\xi_0) = (L_\mathcal{C} \circ T_\vartheta)(\xi_0)$, i.e. the standard likelihood pulled back to $\mathbb{R}^{m_\xi}$. The corresponding latent target density is

$$\rho_\vartheta(\xi_0 \mid \mathcal{C}) = \frac{\Lambda_{\vartheta,\mathcal{C}}(\xi_0)\, r(\xi_0)}{Z_\vartheta}, \qquad Z_\vartheta = \int_{\mathbb{R}^{m_\xi}} \Lambda_{\vartheta,\mathcal{C}}(\xi_0)\, r(\xi_0)\, \mathrm{d}\xi_0,$$

where the posterior is recovered by pushing ρ_ϑ through T_ϑ. That is, if $\xi_0 \sim \rho_\vartheta(\cdot \mid \mathcal{C})$, then $f_0 = T_\vartheta(\xi_0)$ has density $\pi_\vartheta(\cdot \mid \mathcal{C})$. We formalise this equivalence in Prop. 2.2, which states that the latent and process targets agree in distribution.

Proposition 2.2. *Suppose that Assumption 2.1 holds and that $0 < Z_\vartheta < \infty$. Then, if $\xi_0 \sim \rho_\vartheta(\cdot \mid \mathcal{C})$, the random variable $f_0 = T_\vartheta(\xi_0)$ has the conditioned process density $\pi_\vartheta(\cdot \mid \mathcal{C})$.*

2.1 Examples of latent innovation representations

Below we provide three stochastic processes that satisfy Assumption 2.1. Further examples, including max-stable processes, state-space models, and neural processes, are given in App. A.

Gaussian process. Let $f \sim \mathrm{GP}(\mu_\vartheta, k_\vartheta)$, and let $f_0 = (f(s_1), \ldots, f(s_m)) \in \mathbb{R}^m$ denote its values at inputs $s_{1:m}$. Thus $f_0 \sim \mathcal{N}(m_\vartheta, K_\vartheta)$, where $m_\vartheta = (\mu_\vartheta(s_1), \ldots, \mu_\vartheta(s_m))$ and $(K_\vartheta)_{ij} = k_\vartheta(s_i, s_j)$. Let L_ϑ be a matrix satisfying $L_\vartheta L_\vartheta^\top = K_\vartheta$. We can then write

$$f_0 = T_\vartheta(\xi_0) = m_\vartheta + L_\vartheta \xi_0, \qquad \xi_0 \sim \mathcal{N}(0, I_m).$$

Thus, the latent innovation is the whitened Gaussian vector ξ_0, and the generator T_ϑ is the usual affine GP whitening map.

Student-t process. Let $f \sim \mathrm{TP}_\nu(\mu_\vartheta, k_\vartheta)$, where k_ϑ denotes the Student-t scale kernel, and let $f_0 = (f(s_1), \ldots, f(s_m)) \in \mathbb{R}^m$ denote its values at inputs $s_{1:m}$. We then have

$$f_0 = m_\vartheta + \omega^{-1/2} L_\vartheta v,$$

where $v \sim \mathcal{N}(0, I_m)$ and $\omega \sim \mathrm{Gamma}(\nu/2, \nu/2)$ independently. We can also express f_0 in terms of a standard Gaussian reference. Let $a \sim \mathcal{N}(0, 1)$, and define $G_\nu(a) = F^{-1}_{\Gamma,\nu}(\Phi_\mathrm{N}(a))$, where $F_{\Gamma,\nu}$ is the $\mathrm{Gamma}(\nu/2, \nu/2)$ CDF, and Φ_N is the standard normal CDF. Then

$$f_0 = T_\vartheta(\xi_0) = m_\vartheta + G_\nu(a)^{-1/2} L_\vartheta v, \qquad \xi_0 = (v, a) \sim \mathcal{N}(0, I_{m+1}).$$

SDE-driven process. Consider a process defined via the stochastic differential equation

$$\mathrm{d}x_\tau = b_\vartheta(\tau, x_\tau)\,\mathrm{d}\tau + \Sigma_\vartheta(\tau, x_\tau)\,\mathrm{d}w_\tau,$$

where $w = (w_\tau)_{\tau \geq 0}$ is a standard Brownian motion. Define a time grid $0 = \tau_0 < \tau_1 < \cdots < \tau_N = T$, with $\Delta_n = \tau_{n+1} - \tau_n$. The Euler–Maruyama (EM) discretisation of this SDE is

$$x_{n+1} = x_n + b_\vartheta(\tau_n, x_n)\Delta_n + \Sigma_\vartheta(\tau_n, x_n)\sqrt{\Delta_n}\eta_n,$$

where $\eta_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. If x_0 is random, we also write $x_0 = \chi_\vartheta(\eta_{-1})$, with $\eta_{-1} \sim \mathcal{N}(0, I)$, omitting η_{-1} when x_0 is fixed. The latent innovation is then the collection of Gaussian variables $\xi_0 = (\eta_{-1}, \eta_0, \ldots, \eta_{N-1})$, with the obvious modification for deterministic x_0. Meanwhile, the generator T_ϑ is the numerical solver: it takes ξ_0, applies the EM recursion, and returns $(x_0, x_1, \ldots, x_N) = T_\vartheta(\xi_0)$.

3 LatentFlow: Inference with Guided Latent Flows

Whenever the latent likelihood is differentiable, or its score admits a tractable estimator, we can use a guided diffusion to draw samples $\xi_0 \sim \rho_\vartheta(\cdot \mid \mathcal{C})$. We call the resulting framework LatentFlow. The statements provided here are formal; see App. C for a more rigorous treatment.

Noising bridge, potential, and guidance field. We first define a Gaussian noising process on the latent space. Let $\xi_0 \sim r$ and $\varepsilon \sim \mathcal{N}(0, I_{m_\xi})$ be independent of ξ_0. The variance-preserving (VP) latent noising bridge is given by

$$\xi_t = \alpha(t)\xi_0 + \sigma(t)\varepsilon, \qquad \alpha(t)^2 + \sigma(t)^2 = 1, \tag{2}$$

where $\alpha(t) = \exp\{-\frac{1}{2}\int_0^t \beta(s)\,\mathrm{d}s\}$ for some $\beta : [0,1) \to (0,\infty)$ such that $\lim_{t\uparrow 1}\alpha(t) = 0$. Thus ξ_0 is the clean latent variable and ξ_t for $t \uparrow 1$ is increasingly noised.

Since $r(\xi_0) = \mathcal{N}(\xi_0; 0, I_{m_\xi})$, we have $\xi_t \sim r$ for every $t \in [0,1)$. Moreover, the conditional density of $\xi_0 \mid \xi_t$ is given by

$$r_{0|t}(\xi_0) \coloneqq p(\xi_0 \mid \xi_t = u) = \mathcal{N}\big(\xi_0; \alpha(t)u, \sigma(t)^2 I_{m_\xi}\big).$$

The additional condition enters through the latent likelihood Λ. For $t \in [0,1)$, we define the *latent conditioning potential* as

$$\psi(t,u) = \mathbb{E}_{\xi_0 \sim r}\left[\Lambda_{\vartheta,\mathcal{C}}(\xi_0) \mid \xi_t = u\right] = \mathbb{E}_{\varepsilon \sim r}\left[\Lambda_{\vartheta,\mathcal{C}}\left(\alpha(t)u + \sigma(t)\varepsilon\right)\right].$$

This can be viewed as a smoothed version of the latent likelihood. Finally, suppose $\psi(t,u) > 0$ and that $\psi(t,\cdot)$ is differentiable at u. We define for $t \in [0,1)$ the *latent guidance field*

$$g(t,u) = \nabla_u \log \psi(t,u)$$

This will later be used to steer the dynamics in the direction of the increased expected future likelihood.

Exact latent dynamics. Our goal is a sampling procedure for $\xi_0 \sim \rho_0$ with $\rho_0 \equiv \rho_\vartheta$. Let $\xi_t \sim \rho_t$ be the random variable obtained by drawing $\xi_0 \sim \rho_0$ and then applying the VP noising bridge in (2). The family of guided densities $(\rho_t)_{0 \leq t < 1}$ are the one-time marginals of the VP-SDE

$$\mathrm{d}\xi_t = -\tfrac{1}{2}\beta(t)\,\xi_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}w_t, \qquad \xi_0 \sim \rho_0, \quad t \in [0,1),$$

where w is a standard $\mathbb{R}^{m_\xi}$-valued Brownian motion (e.g., Song et al., 2021). By Anderson (1982), the time reversal of these dynamics is also an SDE, given by

$$\mathrm{d}\xi_t = \beta(t)\left[\tfrac{1}{2}\xi_t - g(t,\xi_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{w}_t, \tag{3}$$

where $\overline{w}$ is a reverse-time Brownian motion, and this equation is integrated backwards in t. In particular, if (3) is initialised at $\xi_{t_1} \sim \rho_{t_1}$ and integrated backwards from t_1 to 0, then $\xi_t \sim \rho_t$ for every $t \in [0, t_1]$, and hence $\xi_0 \sim \rho_0$. Since $\rho_{t_1} \Rightarrow \mathcal{N}(0, I_{m_\xi})$ as $t_1 \uparrow 1$, initialising at $\mathcal{N}(0, I_{m_\xi})$ in place of ρ_{t_1} is asymptotically exact, yielding a practical sampler.

The same family of marginal densities can also be generated by a deterministic process known as the probability-flow ODE (see Song et al., 2021). In our case, however, the SDE is often a preferable generating mechanism; see App. C for further details.

Monte Carlo guidance. To generate ξ_0 using the reverse-time SDE in (3) we first need $g(t,u) = \nabla_u \log \psi(t,u)$. Typically, neither $g(t,u)$ nor $\psi(t,u)$ are available in closed form. We can, however, obtain estimates of both via a Monte Carlo estimator. For a given pair (t,u), draw $\varepsilon^{(s)} \sim \mathcal{N}(0, I_{m_\xi})$ and set $\xi_{0|t}^{(s)} = \alpha(t)u + \sigma(t)\varepsilon^{(s)}$ for $s = 1, \dots, S$. Then $\widehat{\psi}_S(t,u) = \frac{1}{S}\sum_{s=1}^{S} \Lambda_{\vartheta,\mathcal{C}}(\xi_{0|t}^{(s)})$ is an unbiased estimator of $\psi(t,u)$. If $\Lambda_{\vartheta,\mathcal{C}}$ is differentiable, then

$$\nabla_u \log \psi(t,u) = \alpha(t) \frac{\mathbb{E}_{\varepsilon\sim r}\left[\nabla_\xi \Lambda_{\vartheta,\mathcal{C}}\big(\alpha(t)u + \sigma(t)\varepsilon\big)\right]}{\mathbb{E}_{\varepsilon\sim r}\left[\Lambda_{\vartheta,\mathcal{C}}\big(\alpha(t)u + \sigma(t)\varepsilon\big)\right]}.$$

When $\Lambda_{\vartheta,\mathcal{C}} > 0$, this motivates the self-normalised estimator

$$\widehat{g}_S(t,u) = \alpha(t) \sum_{s=1}^{S} \bar{w}_s \nabla_\xi \log \Lambda_{\vartheta,\mathcal{C}}(\xi_{0|t}^{(s)}), \qquad \bar{w}_s = \frac{\Lambda_{\vartheta,\mathcal{C}}(\xi_{0|t}^{(s)})}{\sum_{\ell=1}^{S} \Lambda_{\vartheta,\mathcal{C}}(\xi_{0|t}^{(\ell)})}. \tag{4}$$

Since $\Lambda_{\vartheta,\mathcal{C}} = L_{\mathcal{C}} \circ T_\vartheta$, the latent score can be computed by differentiating through the process generator; in practice, this is typically available via automatic differentiation. In particular, when T_ϑ and $L_{\mathcal{C}}$ are differentiable and $L_{\mathcal{C}} > 0$, the chain rule gives

$$\nabla_\xi \log \Lambda_{\vartheta,\mathcal{C}}(\xi) = [D_\xi T_\vartheta(\xi)]^\top \nabla_{f_0} \log L_{\mathcal{C}}(f_0)\big|_{f_0 = T_\vartheta(\xi)},$$

where $D_\xi T_\vartheta(\xi)$ is the Jacobian of $T_\vartheta(\xi)$. For hard constraints, discontinuous likelihoods, or non-differentiable simulators, this gradient-based form must be replaced by a smooth relaxation, a differentiable surrogate, or a derivative-free guidance estimator; see Sec. D.7.

In practice, we replace $g(t,u)$ by $\widehat{g}_S(t,u)$ and solve (3) numerically. The resulting error decomposes into three terms: terminal initialisation, guidance approximation, and numerical discretisation; see App. D. Under a common exact terminal coupling, if the Monte Carlo guidance field is uniformly root-S consistent and the SDE solver has strong order q_{sde}, then the coupled latent grid error is $\mathcal{O}_{\mathbb{P}}(S^{-1/2} + h^{q_{\mathrm{sde}}})$. The error contribution of each can be made arbitrarily small, so the sampling error is limited only by available compute. These bounds pass to process space whenever T_ϑ is Lipschitz.

Bayesian inference of model parameters. Thus far, we have treated the model parameters ϑ as fixed. In practice, unknown process, observation, or simulator parameters can be sampled jointly with the process by augmenting the latent space to include the random variables used to generate their prior. Let p_Θ be the prior density of $\vartheta \in \Theta$ with a tractable generative map $H : \mathbb{R}^{m_\vartheta} \to \Theta$ such that $H(\zeta_0)$ has density p_Θ whenever $\zeta_0 \sim \mathcal{N}(0, I_{m_\vartheta})$. We then augment the latent variable and generator with $\tilde{\xi}_0 := (\xi_0, \zeta_0)$, and $\tilde{T}(\tilde{\xi}_0) := \big(T_{H(\zeta_0)}(\xi_0), H(\zeta_0)\big)$. The LatentFlow construction now applies verbatim to $\tilde{\xi}_0$ in place of ξ_0. In particular, $\tilde{T}(\tilde{\xi}_0) = (f_0, \vartheta)$ has joint posterior density $\pi(f_0, \vartheta \mid \mathcal{C}) \propto L_{\mathcal{C}}(f_0, \vartheta) p_\vartheta(f_0) p_\Theta(\vartheta)$. Thus any finite collection of learnable model parameters admitting a tractable prior generator can be included in the latent state and sampled under a fully Bayesian joint posterior using LatentFlow. Further details are provided in App. E.

4 Related work

The closest antecedent to our work is FlowGP (Moss et al., 2026). Our framework contains FlowGP as a special case, but also applies to a substantially broader class of stochastic processes; see App. A. More broadly, our approach relates to diffusion models, flow matching, and stochastic interpolants, which generate samples via continuous-time transport (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Albergo et al., 2025). Unlike pretrained diffusion-model guidance (e.g., Ho & Salimans, 2022), however, LatentFlow does not need to be pre-trained on a neural estimate of an unconditional data score or reverse process. The noising bridge is imposed in innovation space, and the guidance is the gradient of a smoothed pulled-back likelihood under the known latent bridge.

A closely related line of work conditions generative models by performing inference in the latent noise space (Whang et al., 2021; Holden et al., 2022; Patel et al., 2022; Achituve et al., 2025; Askari et al., 2025; Purohit et al., 2025; Venkatraman et al., 2025; Xia et al., 2026). These methods share the idea of conditioning a generator by changing the distribution of its input noise, but focus only on learned neural generators. Our framework is significantly more general: it includes pre-trained neural generators as a special case, but also extends to a much broader class of genuine stochastic-process priors. The papers above also leverage very different, often more expensive, samplers, for instance, Langevin dynamics (Purohit et al., 2025), HMC (Patel et al., 2022; Xia et al., 2026), pCN (Holden et al., 2022), or SMC (Achituve et al., 2025). LatentFlow instead targets the latent posterior using a guided, training-free diffusion, whose guidance field is constructed from the pulled-back likelihood.

Our work is also related to many other areas, including diffusion bridges, rare-event simulation, and transport-assisted MCMC; see App. F for a more detailed discussion.

5 Numerical Experiments

We now present a range of numerical experiments designed to assess the flexibility, accuracy, and scalability of LatentFlow. We first verify the effectiveness of LatentFlow on three standard sampling benchmarks. In each, it accurately samples the true posterior, and outperforms several standard sampling benchmarks; see App. G.1 for full details. Below, we consider several higher-dimensional and more challenging examples. All timings are reported on a 13th Gen Intel i9-13900K CPU.

Spatial processes. We first consider spatial processes, which are central to many scientific applications including environmental modelling and ecology (e.g., Cressie, 1993). While GPs provide a convenient spatial prior, many applications exhibit extremal or heavy tails, discontinuities, or sharp local anomalies, motivating alternatives such as Student-t and Lévy processes (Wolpert et al., 2011; Shah et al., 2014). The difficulty of conditioning these is twofold. First, the prior can break conjugacy as non-Gaussian priors induce posteriors involving scale mixtures or jump-like innovations (Walchessen et al., 2025). Second, non-linear conditions such as areal maxima or censored measurements do not admit closed-form posteriors, even under a Gaussian prior. In Fig. 2, we use LatentFlow to condition three spatial priors of increasing non-Gaussianity on four conditions; see also Figs. 10 to 12 in App. G.2. In all cases, the resulting samples respect the imposed conditioning information, whilst also preserving the distinctive behaviour of each prior. The speed of LatentFlow is notable here: conditional samples are generated in just $3-4$ seconds, significantly faster than alternative neural-network based methods (e.g., Zammit-Mangion et al., 2025).

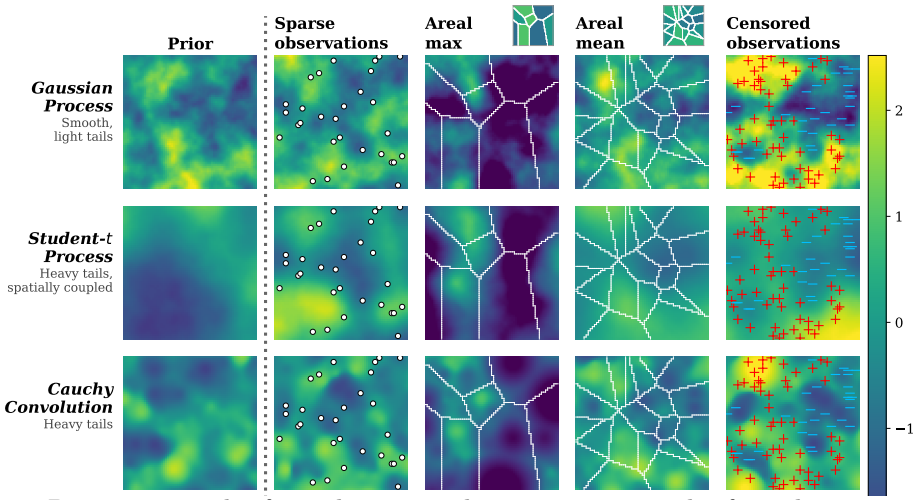
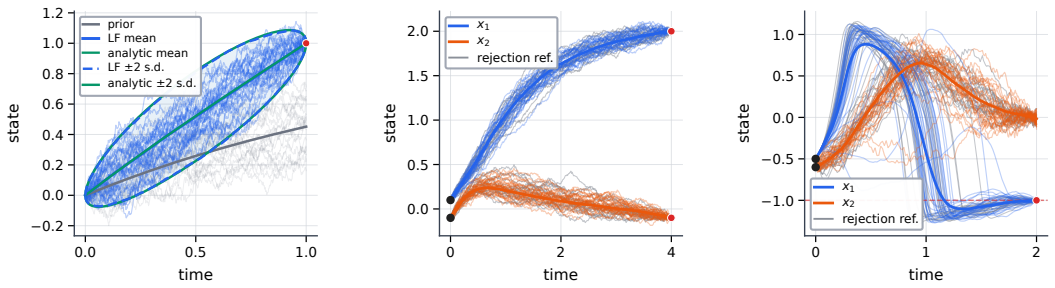


Figure 2: Posterior samples from three spatial process priors under four observation types via LatentFlow. **Rows:** Gaussian process (Matérn-$\frac{3}{2}$); Student-t process ($\nu = 4$); Cauchy convolution field. **Columns:** unconditional; 30 point observations; areal maxima (8 regions); areal means (20 regions); 60 censored observations (+/−). Thumbnails show areal structures. Sampling time: $3-4$s per sample.



(a) Ornstein–Uhlenbeck. (b) Cell differentiation. (c) FitzHugh–Nagumo.

Figure 3: Diffusion bridges generated by LatentFlow. **(a)** An Ornstein–Uhlenbeck bridge. **(b)** A two-dimensional nonlinear cell-differentiation diffusion conditioned to reach a prescribed terminal state. **(c)** A FitzHugh–Nagumo diffusion conditioned on a rare transition between metastable states. Red markers indicate terminal targets, blue & orange curves indicate conditioned state components; grey curves indicate prior or rejection-reference trajectories.

Temporal processes. We next consider temporal processes, which include diffusion processes (e.g., Øksendal, 2003) and dynamical systems with random initial conditions or exogenous forcing (e.g., Law et al., 2015). This setting presents a different use case for LatentFlow: the object being conditioned is no longer a static field, but a trajectory generated by propagating latent randomness through a dynamical model.

Diffusion bridges. We first consider the special case of diffusion bridges—diffusion processes conditioned on a fixed terminal condition—which arise in fields ranging from computational chemistry (Bolhuis et al., 2002) to developmental dynamics (Wang et al., 2011). Except in a few tractable cases, such bridges depend on an unavailable transition density, motivating bespoke guided proposals and path-space MCMC (Delyon & Hu, 2006; Schauer et al., 2017), as well as neural approaches that approximate the bridge score or guidance (e.g., Baker et al., 2024; Heng et al., 2025; Yang et al., 2025b). In Fig. 3, we instead use LatentFlow for this problem, approximating exact endpoint conditioning by a narrow finite-width likelihood. Our results indicate that LatentFlow is capable of generating samples which agree closely with the true posterior in examples where the ground truth can be obtained in closed form (Fig. 3a; Figs. 13 and 14 in Sec. G.3), and are consistent with rejection-sampled references in several more difficult non-linear examples (Figs. 3b and 3c; Figs. 15 and 17 in App. G.3).

Trajectory-level conditioning. We next consider more general trajectory-level conditioning, where the information may involve observations, terminal events, or pathwise threshold constraints (e.g., Bolhuis & Swenson, 2021; Grafke, 2025). Such constraints act on the realised trajectory after the latent randomness has been propagated through a possibly nonlinear dynamical simulator, and need not in general reduce to a standard filtering or smoothing problem. In Fig. 4, we consider Lotka–Volterra dynamics (Lotka, 1925; Volterra, 1926), an SIR model (Kermack & McKendrick, 1927), and the Heston stochastic-volatility model (Heston, 1993). The resulting conditional paths satisfy the imposed trajectory-level information while retaining the characteristic dynamics of each model. The speed is again notable here; sampling takes just 1s for SIR and Lotka–Volterra, and 3s for Heston.

Spatio-Temporal Processes Finally, we consider spatio-temporal models, which describe the random evolution of spatial fields. Such models arise in applications such as turbulent transport, environmental dispersion, materials science, excitable media, and phase separation (Majda & Kramer, 1999; Allen & Cahn, 1979; Nagumo et al., 1962). Conditioning them on sparse observations, future events, or aggregate space–time constraints is central to data assimilation, inverse modelling, and rare-event prediction (Dashti & Stuart, 2017; Cérou & Guyader, 2007; Cérou et al., 2012; Botev & L'Ecuyer, 2020). We consider three examples: a stochastic Allen–Cahn field for noisy phase separation, an advection–diffusion plume model, and a stochastic FitzHugh–Nagumo excitable-media model that produces travelling activation waves (FitzHugh, 1961; Nagumo et al., 1962; Allen & Cahn, 1979; Majda & Kramer, 1999). Representative samples are shown in Fig. 5. Across all three models, LatentFlow produces fields that remain consistent with the prior while satisfying the imposed conditions.

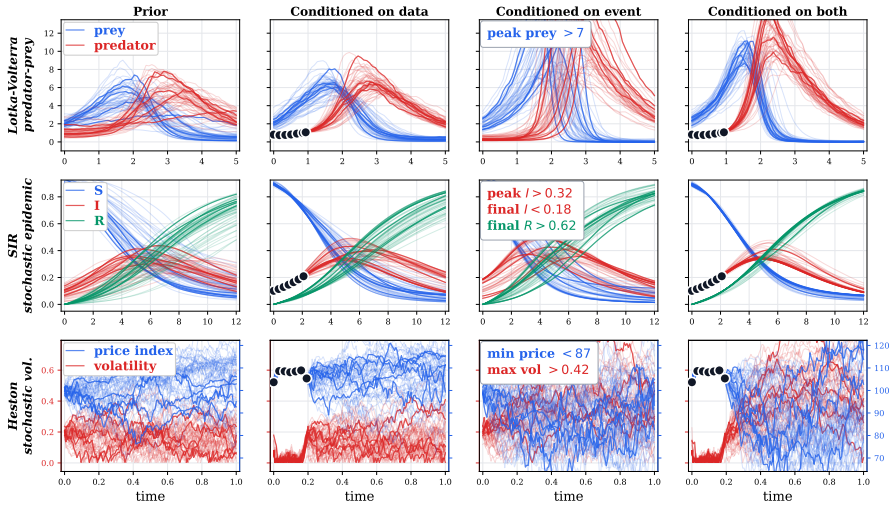


Figure 4: Posterior samples from three SDE priors conditioned on data and rare events using LatentFlow. **Rows**: Lotka–Volterra predator–prey dynamics, a stochastic SIR epidemic, and Heston stochastic volatility. **Columns**: the prior, the posterior given observations (black circles), the posterior given rare events: a prey boom, a fast growing but fast terminating epidemic, and a stress event (a market crash followed by a volatility spike). Sampling time: $<$ 1s for SIR and Lotka–Volterra; 3s Heston.

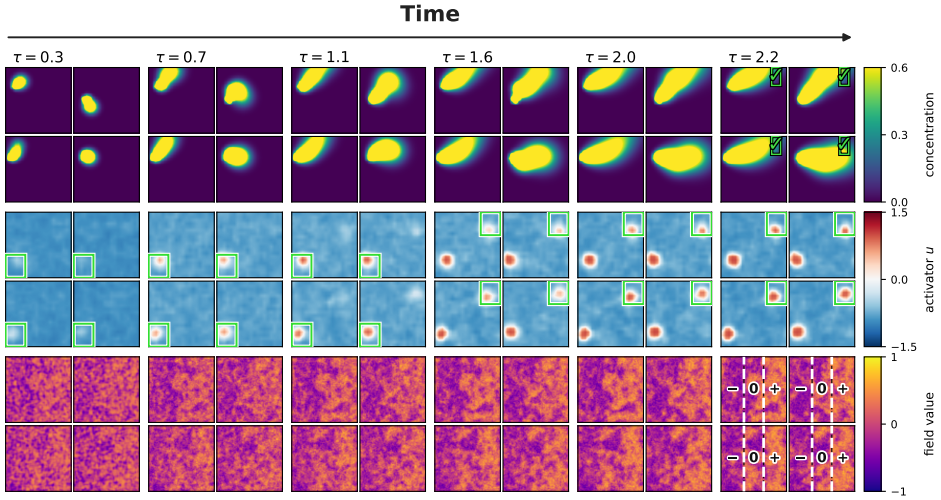


Figure 5: Posterior samples from three spatio-temporal systems using LatentFlow. **Top**: a concentration plume conditioned on triggering a downstream detector evolving under advection–diffusion dynamics (160 latent dimensions). **Middle**: a FitzHugh–Nagumo excitable medium conditioned on sequential pacemaker firing (40K latent dimensions). **Bottom**: an Allen–Cahn phase field conditioned on a prescribed phase front (122K latent dimensions). Each row shows four independent draws, over six time snapshots. Sampling time: 1s for the advection–diffusion; 5s for FitzHugh–Nagumo, and 2s for Allen–Cahn.

Bayesian inference of model parameters. We conclude, following Sec. 3, by using LatentFlow for joint parameter and process inference, augmenting the latent state with the parameters' prior generators so the process and its unknown parameters are sampled together. We revisit the Cauchy-convolution field and the stochastic Lotka–Volterra system, placing a prior on the kernel lengthscale and the four predator–prey rates respectively. The results are shown in Fig. 6: each row gives the ground truth alongside prior and guided-posterior samples, and the induced parameter posteriors. Where the prior draws ignore the conditioning information, the posterior draws respect the observations and constraints, and the parameter posteriors concentrate around the ground truth.

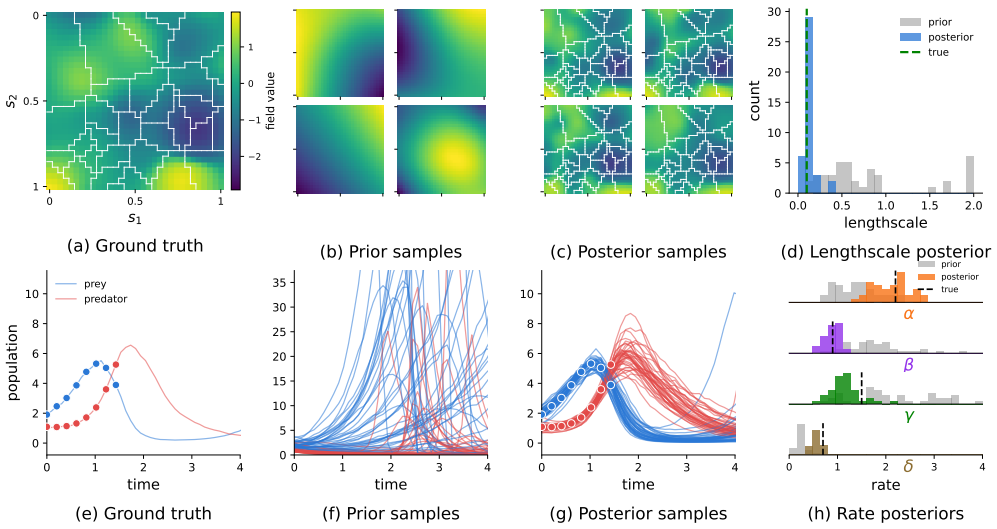


Figure 6: Joint Bayesian parameter and process inference with LatentFlow. **Top**: Cauchy-convolution field, observed only via the *maximum* value within each of $K = 32$ irregular regions. (a) Truth. (b) Samples from the prior. (c) Guided posterior samples respecting the regional-maximum condition. (d) Lengthscale prior vs. posterior; the posterior concentrates on the true value ($\ell = 0.1$, dashed). **Bottom**: stochastic Lotka–Volterra process, with dense early observations (dots). (e) Truth. (f) Samples from the prior. (g) Guided posterior samples respecting the early-window observations. (h) Prior vs. posterior for the four rates.

In this case, to assess the quality of the posterior samples generated by LatentFlow, we compare against two MCMC reference samplers, which we run in our augmented latent space: preconditioned Crank–Nicolson (pCN) (Cotter et al., 2013) and elliptical slice sampling (ESS) (Murray et al., 2010). For Lotka–Volterra, we also compare with the gold-standard No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014), which can be run in the native parameter space; see App. H for further details. For Lotka–Volterra, all schemes converge to very similar posteriors; see Tab. 17 in App. H. In particular, all four samplers recover marginals for α, β which concentrate around the "true" (i.e., data generating) parameter and, as expected, leave γ, δ weakly identified (these parameters are not constrained by early time conditions). These results provide further evidence in a more challenging setting that LatentFlow faithfully samples the true posterior. By contrast, for the Cauchy example, both MCMC samplers fail to mix (see App. H for details). This is likely due to the areal-maximum likelihood inducing a rough, multimodal latent posterior that even dimension-robust Gaussian-prior MCMC cannot traverse. In contrast, LatentFlow successfully draws independent samples from the lengthscale posterior, concentrating on the true lengthscale (see Fig. 6).

6 Discussion

LatentFlow replaces the bespoke samplers that have typically been required for conditioning stochastic processes with a single, unified approach. The central idea is to move conditioning from process space to a latent innovation space, with the likelihood pulled back through the process generator and the resulting conditional law sampled using guided reverse-time dynamics. This gives an exact target-law construction, with practical approximation confined to three explicit and systematically reducible sources: finite terminal noising, Monte Carlo guidance, and time discretisation. This is a fundamental shift in capability.

Several limitations remain. LatentFlow requires a *tractable* latent innovation representation of the discretised prior, and its performance can depend on the dimension, smoothness, and geometry of this representation. It also requires pointwise evaluation of the pulled-back likelihood, and ideally that the likelihood is differentiable through the generator. Discontinuous simulators, hard constraints, multimodal conditionals, or poorly scaled latent parameterisations may therefore require relaxations, surrogates, derivative-free guidance, or larger Monte Carlo budgets.

AI use statement

In this work, we used generative AI tools to generate synthetic data sets, formulate mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, and implement methods. We have not used generative AI tools to provide critical ingredients for proving mathematical claims, help develop theoretical models or conceptual frameworks, to propose or refine hypotheses, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or to interpret results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Idan Achituve, Hai Victor Habi, Amir Rosenfeld, Arnon Netzer, Idit Diamant, and Ethan Fetaya. Inverse problem sampling in latent space using sequential Monte Carlo. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 420–443, 2025.

Christian Agrell. Gaussian processes with linear operator inequality constraints. *Journal of Machine Learning Research*, 20(135):1–36, 2019.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

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.

Samuel M. Allen and John W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

Mauricio Álvarez, David Luengo, and Neil D. Lawrence. Latent force models. In *Proceedings of The 12th International Conference on Artificial Intelligence and Statistics*, pp. 9–16, 2009.

Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (11):2693–2705, 2013.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1–2):5–43, 2003.

Michael Arbel, Alexander G. D. G. Matthews, and Arnaud Doucet. Annealed flow transport Monte Carlo. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 318–330, 2021.

Hossein Askari, Yadan Luo, Hongfu Sun, and Fred Roosta. Latent refinement via flow matching for training-free linear inverse problem solving. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Lachlan Astfalck, Deborshee Sen, Sayan Patra, Edward Cripps, and David Dunson. Posterior projection for inference in constrained spaces. *arXiv preprint arXiv:1812.05741*, 2018.

Lachlan Astfalck, Cassandra Bird, and Daniel Williamson. Generalised Bayes linear inference. *arXiv preprint arXiv:2405.14145*, 2024.

Elizabeth L. Baker, Alexander Denker, and Jes Frellsen. Supervised guidance training for infinite-dimensional diffusion models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Elizabeth Louise Baker, Gefan Yang, Michael Lind Severinsen, Christy Anna Hipsley, and Stefan Sommer. Conditioning non-linear and infinite-dimensional diffusion processes. In *Advances in Neural Information Processing Systems*, volume 37, pp. 10801–10826, 2024.

Elizabeth Louise Baker, Moritz Schauer, and Stefan Sommer. Score matching for bridges without learning time-reversals. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, pp. 775–783, 2025.

Lorenzo Baldassari, Ali Siahkoohi, Josselin Garnier, Knut Sølna, and Maarten V. de Hoop. Conditional score-based diffusion models for Bayesian inference in infinite dimensions. In *Advances in Neural Information Processing Systems*, volume 36, pp. 24262–24290, 2023.

Lorenzo Baldassari, Ali Siahkoohi, Josselin Garnier, Knut Sølna, and Maarten V. de Hoop. Taming score-based diffusion priors for infinite-dimensional nonlinear inverse problems. *arXiv preprint arXiv:2405.15676*, 2024.

Lorenzo Baldassari, Josselin Garnier, Knut Sølna, and Maarten V. de Hoop. Preconditioned Langevin dynamics with score-based generative models for infinite-dimensional linear Bayesian inverse problems. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E. Jacob. Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*, 2019.

Alexandros Beskos, Frank J. Pinski, Jesus Maria Sanz-Serna, and Andrew M. Stuart. Hybrid Monte Carlo on hilbert spaces. *Stochastic Processes and their Applications*, 121(10): 2201–2230, 2011.

Joris Bierkens, Frank van der Meulen, and Moritz Schauer. Simulation of elliptic and hypo-elliptic conditional diffusions. *Advances in Applied Probability*, 52(1):173–212, 2020.

Mogens Bladt and Michael Sørensen. Simple simulation of diffusion bridges with application to likelihood inference for diffusions. *Bernoulli*, 20(2):645–675, 2014.

Mogens Bladt, Samuel Thomas John Finch, and Michael Sørensen. Simulation of multivariate diffusion bridges. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(2):343–369, 2016.

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

Vanessa Böhm, François Lanusse, and Uroš Seljak. Uncertainty quantification with generative models. *arXiv preprint arXiv:1910.10046*, 2019.

Peter G. Bolhuis and David W. H. Swenson. Transition path sampling as Markov chain Monte Carlo of trajectories: Recent algorithms, software, applications, and future outlook. *Advanced Theory and Simulations*, 4(4):2000237, 2021.

Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53:291–318, 2002.

Zdravko I. Botev and Pierre L'Ecuyer. Sampling conditionally on a rare event via generalized splitting. *INFORMS Journal on Computing*, 32(4):986–995, 2020.

Alberto Cabezas and Christopher Nemeth. Transport elliptical slice sampling. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 3664–3676, 2023.

Alberto Cabezas, Louis Sharrock, and Christopher Nemeth. Markovian flow matching: Accelerating MCMC with continuous normalizing flows. In *Advances in Neural Information Processing Systems*, volume 37, pp. 104383–104411, 2024.

Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte Carlo guided denoising diffusion models for Bayesian linear inverse problems. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007.

Frédéric Cérou, Pierre Del Moral, Teddy Furon, and Arnaud Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808, 2012.

Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447: 110668, 2021.

Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169(2):671–691, 2016.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Adrien Corenflos, Zheng Zhao, Thomas B. Schön, Simo Särkkä, and Jens Sjölund. Conditioning diffusion models by explicit forward-backward bridging. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, pp. 3709–3717, 2025.

Simon L. Cotter, Gareth O. Roberts, Andrew M. Stuart, and David White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28(3): 424–446, 2013.

Noel A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, New York, revised edition, 1993. ISBN 9780471002550.

Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. *Annales de la Faculté des Sciences de Toulouse: Mathématiques*, 21(3): 529–555, 2012.

Andreas Damianou and Neil D. Lawrence. Deep Gaussian processes. In *Proceedings of The 16th International Conference on Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Masoumeh Dashti and Andrew M. Stuart. The Bayesian approach to inverse problems. In Roger Ghanem, David Higdon, and Houman Owhadi (eds.), *Handbook of Uncertainty Quantification*, pp. 311–428. Springer, Cham, 2017.

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, pp. 17695–17709, 2021.

Pierre Del Moral and Josselin Garnier. Genealogical particle analysis of rare events. *The Annals of Applied Probability*, 15(4):2496–2534, 2005.

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.

Christoph Dellago, Peter G. Bolhuis, Felix S. Csajka, and David Chandler. Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics*, 108(5): 1964–1977, 1998.

Christoph Dellago, Peter G. Bolhuis, and Phillip L. Geissler. Transition path sampling. In Ilya Prigogine and Stuart A. Rice (eds.), *Advances in Chemical Physics*, volume 123, pp. 1–78. John Wiley & Sons, 2002.

Bernard Delyon and Ying Hu. Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications*, 116(11):1660–1675, 2006.

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

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Kieran Didi, Francisco Vargas, Simon V. Mathis, Vincent Dutordoir, Emile Mathieu, Urszula J. Komorowska, and Pietro Liò. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design. In *NeurIPS 2023 Workshop AI4D3*, 2023.

Joseph L. Doob. Conditional Brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957.

Arnaud Doucet, Nando de Freitas, and Neil Gordon (eds.). *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer, New York, NY, 2001. ISBN 978-0-387-95146-1.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 8990–9012, 2023.

Tarek A. El Moselhy and Youssef M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, 2012.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 16993–17029, 2025.

Marc Anton Finzi, Anudhyan Boral, Andrew Gordon Wilson, Fei Sha, and Leonardo Zepeda-Núñez. User-defined event sampling and uncertainty quantification in diffusion models for physical dynamical systems. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10136–10152, 2023.

Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

Hans Föllmer. An entropy approach to the time reversal of diffusion processes. In Michel Métivier and Étienne Pardoux (eds.), *Stochastic Differential Systems: Filtering and Control*, volume 69 of *Lecture Notes in Control and Information Sciences*, pp. 156–163. Springer, Berlin, Heidelberg, 1985.

Giulio Franzese, Giulio Corallo, Simone Rossi, Markus Heinonen, Maurizio Filippone, and Pietro Michiardi. Continuous-time functional diffusion processes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37370–37400, 2023.

Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl Edward Rasmussen. Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems*, volume 26, pp. 3156–3164, 2013.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems*, volume 27, pp. 3680–3688, 2014.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Andrew Golightly and Darren J. Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3): 1674–1693, 2008.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Tobias Grafke. Sampling conditioned diffusions via pathspace projected monte carlo. *arXiv preprint arXiv:2506.15743*, 2025.

Matthew Graham and Amos Storkey. Asymptotically exact inference in differentiable generative models. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 499–508, 2017.

Matthew M. Graham, Alexandre H. Thiery, and Alexandros Beskos. Manifold Markov chain Monte Carlo methods for Bayesian inference in diffusion models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(4):1229–1256, 2022.

Mamikon Gulian, Ari L. Frankel, and Laura P. Swiler. Gaussian process regression constrained by boundary value problems. *Computer Methods in Applied Mechanics and Engineering*, 388:114117, 2022.

Zhengyi Guo, Wenpin Tang, and Renyuan Xu. Conditional diffusion guidance under hard constraint: A stochastic analysis approach. *arXiv preprint arXiv:2602.05533*, 2026.

Martin Hairer, Andrew M. Stuart, and Sebastian J. Vollmer. Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 24(6): 2455–2490, 2014.

Oliver Hamelijnck, Arno Solin, and Theodoros Damoulas. Physics-informed variational state-space Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 37, pp. 98505–98536, 2024.

Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching. *Biometrika*, 112(4):asaf048, 2025.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *Proceedings of The 18th International Conference on Artificial Intelligence and Statistics*, pp. 351–360, 2015.

Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47): 1593–1623, 2014.

Matthew D. Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. NeuTra-lizing bad geometry in hamiltonian Monte Carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

Matthew Holden, Marcelo Pereyra, and Konstantinos C. Zygalakis. Bayesian imaging with data-driven priors encoded by neural networks: Theory, methods, and algorithms. *SIAM Journal on Imaging Sciences*, 15(2):892–924, 2022.

Bamdad Hosseini, Alexander W. Hsu, and Amirhossein Taghvaei. Conditional optimal transport on function spaces. *SIAM/ASA Journal on Uncertainty Quantification*, 13(1): 304–338, 2025.

Jian Huang, Yuling Jiao, Lican Kang, Xu Liao, Jin Liu, and Yanyan Liu. Schrödinger–Föllmer sampler: Sampling without ergodicity. *arXiv preprint arXiv:2106.10880*, 2021.

Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B. Schön. Linearly constrained Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1215–1224, 2017.

William O. Kermack and Anderson G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721, 1927.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 3934–3942, 2024.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Malte Kuss and Carl Edward Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

Markus Lange-Hegermann. Algorithmic linearly constrained Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2137–2148, 2018.

Kody J. H. Law, Andrew M. Stuart, and Konstantinos C. Zygalakis. *Data Assimilation: A Mathematical Introduction*, volume 62 of *Texts in Applied Mathematics*. Springer, 2015.

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

Jae Hyun Lim, Nikola B. Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, Christopher Pal, Arash Vahdat, and Anima Anandkumar. Score-based diffusion models in function space. *Journal of Machine Learning Research*, 26(158):1–62, 2025.

Lizhen Lin and David B. Dunson. Bayesian monotone regression using Gaussian process projection. *Biometrika*, 101(2):303–317, 2014.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

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*, pp. 22042–22062, 2023a.

Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos A. Theodorou, and Ricky T. Q. Chen. Generalized Schrödinger bridge matching. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the 11th International Conference on Learning Representations*, 2023b.

Andrés F. López-Lopera, François Bachoc, Nicolas Durrande, and Olivier Roustant. Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

Alfred J. Lotka. *Elements of Physical Biology*. Williams & Wilkins, Baltimore, 1925.

Chao Ma, Yingzhen Li, and Jose Miguel Hernandez-Lobato. Variational implicit processes. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4222–4233, 2019.

Hassan Maatouk and Xavier Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, 2017.

Hassan Maatouk, Didier Rullière, and Xavier Bay. Bayesian analysis of constrained Gaussian processes. *Bayesian Analysis*, 20(3):973–1002, 2025.

Andrew J. Majda and Peter R. Kramer. Simplified models for turbulent diffusion: Theory, numerical modelling, and physical phenomena. *Physics Reports*, 314(4–5):237–574, 1999.

Konstantin Mark, Leonard Galustian, Maximilian P.-P. Kovar, and Esther Heid. Feynman–kac-flow: Inference steering of conditional flow matching to an energy-tilted posterior. *arXiv preprint arXiv:2509.01543*, 2025.

Henry B. Moss, Lachlan Astfalck, Thomas Cowperthwaite, Colin Doumont, Sam Willis, Philipp Hennig, Christopher Nemeth, and Andrew Zammit-Mangion. Conditioning Gaussian processes on almost anything. *arXiv preprint arXiv:2605.21041*, 2026.

Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of The 13th International Conference on Artificial Intelligence and Statistics*, pp. 541–548, 2010.

Jin-Ichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

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

Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

Hannes Nickisch, Arno Solin, and Alexander Grigorevskiy. State space Gaussian processes with non-Gaussian likelihood. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3789–3798, 2018.

Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. MCMC should mix: Learning energy-based model with neural transport latent space MCMC. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer, Berlin, 6 edition, 2003.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Omiros Papaspiliopoulos, Gareth O. Roberts, and Osnat Stramer. Data augmentation for diffusions. *Journal of Computational and Graphical Statistics*, 22(3):665–688, 2013.

Meet Hemant Parikh, Yaqin Chen, and Jian-Xun Wang. D-Flow SGLD: Source-space posterior sampling for scientific inverse problems with flow matching. *arXiv preprint arXiv:2602.21469*, 2026.

Byoungwoo Park, Jungwon Choi, Sungbin Lim, and Juho Lee. Stochastic optimal control for diffusion bridges in function spaces. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28745–28771, 2024.

Matthew D. Parno and Youssef M. Marzouk. Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682, 2018.

Dhruv V. Patel, Deep Ray, and Assad A. Oberai. Solution of physics-based Bayesian inverse problems with deep generative priors. *Computer Methods in Applied Mechanics and Engineering*, 400:115428, 2022.

Stefano Peluchetti. Diffusion bridge mixture transports, Schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

Jakiw Pidstrigach, Elizabeth Louise Baker, Carles Domingo-Enrich, George Deligiannidis, and Nikolas Nüsken. Conditioning diffusions using Malliavin calculus. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 49292–49315, 2025.

Thorben Pieper-Sethmacher, Frank van der Meulen, and Aad van der Vaart. Simulation of infinite-dimensional diffusion bridges. *arXiv preprint arXiv:2503.13177*, 2025.

Vishal Purohit, Matthew Repasky, Jianfeng Lu, Qiang Qiu, Yao Xie, and Xiuyuan Cheng. Consistency posterior sampling for diverse image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28327–28336, 2025.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1177–1184, 2007.

Carl Edward Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. In *Proceedings of The 13th International Conference on Artificial Intelligence and Statistics*, pp. 645–652, 2010.

Gareth O. Roberts and Osnat Stramer. On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm. *Biometrika*, 88(3):603–621, 2001.

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, FL, 2005.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge, 2013.

Moritz Schauer, Frank van der Meulen, and Harry van Zanten. Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli*, 23(4A):2917–2950, 2017.

Kiyoung Seong, Seonghyun Park, Seonghwan Kim, Woo Youn Kim, and Sungsoo Ahn. Transition path sampling with improved off-policy training of diffusion path samplers. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Amar Shah, Andrew Gordon Wilson, and Zoubin Ghahramani. Student-t processes as alternatives to Gaussian processes. In *Proceedings of The 17th International Conference on Artificial Intelligence and Statistics*, pp. 877–885, 2014.

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*, pp. 55810–55827, 2025.

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*, pp. 55906–55949, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

Andrew M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Laura Swiler, Mamikon Gulian, Ari Frankel, Cosmin Safta, and John Jakeman. A survey of constrained Gaussian process Regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2):119–156, 2020.

Kirill Tamogashev and Nikolay Malkin. Data-to-energy stochastic dynamics. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *Proceedings of the 11th International Conference on Learning Representations*, 2023a.

Francisco Vargas, Andrius Ovsianas, David Fernandes, Mark Girolami, Neil D. Lawrence, and Nikolas Nüsken. Bayesian learning via neural Schrödinger–Föllmer flows. *Statistics and Computing*, 33(3), 2023b.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled Monte Carlo diffusions. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Siddarth Venkatraman, Mohsin Hasan, Minsu Kim, Luca Scimeca, Marcin Sendera, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 61212–61239, 2025.

Vito Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

Julia Walchessen, Andrew Zammit-Mangion, Raphaël Huser, and Mikael Kuusela. Neural conditional simulation for complex spatial processes. *arXiv preprint arXiv:2508.20067*, 2025.

Jin Wang, Kun Zhang, Li Xu, and Erkang Wang. Quantifying the waddington landscape and biological paths for development and differentiation. *Proceedings of the National Academy of Sciences*, 108(20):8257–8262, 2011.

Yuanzhe Wang and Alexandre M. Tartakovsky. Latent diffusion posterior sampling with surrogate likelihood guidance for PDE inverse problems. *arXiv preprint arXiv:2606.26592*, 2026.

Zifan Wang, Alice Harting, Matthieu Barreau, Michael M. Zavlanos, and Karl H. Johansson. Source-guided flow matching. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Jay Whang, Erik M. Lindgren, and Alexandros G. Dimakis. Composing normalizing flows for inverse problems. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11158–11169, 2021.

Gavin A. Whitaker, Andrew Golightly, Richard J. Boys, and Christopher G. Sherlock. Bayesian inference for diffusion-driven mixed-effects models. *Bayesian Analysis*, 12(2): 435–463, 2017.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In

Proceedings of the 37th International Conference on Machine Learning, pp. 10292–10302, 2020.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise conditioning of Gaussian processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021.

Robert L. Wolpert, Merlise A. Clyde, and Chong Tu. Stochastic expansions using continuous dictionaries: Lévy adaptive regression kernels. *The Annals of Statistics*, 39(4):1916–1962, 2011.

Chen Henry Wu, Saman Motamed, Shaunak Srivastava, and Fernando D. De la Torre. Generative visual prompt: Unifying distributional control of pre-trained generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22422–22437, 2022.

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, pp. 31372–31403, 2023.

Luhuan Wu, Yi Han, Christian Andersson Naesseth, and John P. Cunningham. Reverse diffusion sequential Monte Carlo samplers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yingzhi Xia, Setthakorn Tanomkiattikun, Liangli Zhen, and Zaiwang Gu. Noise-adaptive diffusion sampling for inverse problems without task-specific tuning. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Gefan Yang, Elizabeth Louise Baker, Michael Lind Severinsen, Christy Anna Hipsley, and Stefan Sommer. Infinite-dimensional diffusion bridge simulation via operator learning. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258, pp. 3556–3564, 2025a.

Gefan Yang, Frank van der Meulen, and Stefan Sommer. Neural guided diffusion bridges. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 71210–71230, 2025b.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided diffusion sampling on function spaces with applications to PDEs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Andrew Zammit-Mangion, Matthew Sainsbury-Dale, and Raphaël Huser. Neural methods for amortized inference. *Annual Review of Statistics and Its Application*, 12(1):311–335, 2025.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Zheng Zhao, Ziwei Luo, Jens Sjölund, and Thomas B. Schön. Conditional sampling within generative diffusion models. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2299):20240329, 2025.

A Examples of latent innovation representations

This appendix contains examples of stochastic-process priors that admit the latent innovation representation in Assumption 2.1, or its measure-theoretic extension; see Remark B.1 for details. Throughout, $s_{1:m}$ denotes the discretisation inputs, $f_0 \in E_m$ denotes the finite-dimensional process realisation, and T_ϑ denotes a generator satisfying $f_0 = T_\vartheta(\xi_0)$.

In several examples, the model is most naturally written in terms of a native innovation variable ζ_0, such as a uniform simulator seed, a Gamma scale variable, or a collection of non-Gaussian increments. To ensure all examples are consistent with the common latent form used by LatentFlow, we express this native innovation as a deterministic function of a standard Gaussian reference: $\zeta_0 = H_\vartheta(\xi_0)$, with $\xi_0 \sim \mathcal{N}(0, I)$. The generator in Assumption 2.1 is then the composite map from the Gaussian innovation to the process realisation. When the native innovation is already Gaussian, H_ϑ is simply the identity map.

Table 1: Examples of stochastic-process priors and their latent innovation representations.

Model class	**Innovation representation**
Gaussian, Student-t, and basis-expansion priors	
Gaussian process	A finite-dimensional realisation $f_0 = (f(s_1), \dots, f(s_m))$ of a Gaussian process can be written as $$f_0 = \mu_\vartheta + L_\vartheta \xi_0, \qquad L_\vartheta L_\vartheta^\top = K_\vartheta,$$ where $(K_\vartheta)_{ij} = k_\vartheta(s_i, s_j)$ is the kernel matrix, and $\xi_0 \sim \mathcal{N}(0, I_m)$. The innovation is given by $$\xi_0 \in \mathbb{R}^m, \qquad \xi_0 \sim \mathcal{N}(0, I_m).$$ The generator is given by $$T_\vartheta(\xi_0) = \mu_\vartheta + L_\vartheta \xi_0.$$ This is the affine whitening map used in finite-dimensional Gaussian process inference.
Student-t process	A finite-dimensional realisation $f_0 = (f(s_1), \dots, f(s_m))$ of a Student-t process can be written as $$f_0 = \mu_\vartheta + \omega^{-1/2} L_\vartheta v, \qquad L_\vartheta L_\vartheta^\top = K_\vartheta,$$ where $(K_\vartheta)_{ij} = k_\vartheta(s_i, s_j)$ is the scale matrix induced by the Student-t process kernel k_ϑ, and v and ω are independent, with $v \sim \mathcal{N}(0, I_m)$ and $\omega \sim \mathrm{Gamma}(\nu/2, \nu/2)$ in the shape–rate parameterisation. The natural innovation is given by $$\zeta_0 = (v, \omega) \in \mathbb{R}^{m+1}.$$ Let $w = \Phi_{\mathrm{N}}^{-1}(F_{\Gamma,\nu}(\omega))$, where $F_{\Gamma,\nu}$ is the distribution function of $\mathrm{Gamma}(\nu/2, \nu/2)$, and Φ_{N} is the standard normal distribution function. The Gaussian innovation is then given by $$\xi_0 = (v, w) \in \mathbb{R}^{m+1}, \qquad \xi_0 \sim \mathcal{N}(0, I_{m+1}).$$ The corresponding generator is given by $$T_\vartheta(\xi_0) = \mu_\vartheta + F_{\Gamma,\nu}^{-1}(\Phi_{\mathrm{N}}(w))^{-1/2} L_\vartheta v.$$ This gives a heavy-tailed analogue of the Gaussian process while retaining an explicit scale-mixture innovation map.

continued on next page

Model class	Innovation representation
Finite-basis, random-feature, or pathwise GP approximation	A truncated Karhunen–Loève expansion, finite basis expansion, random-feature approximation, or pathwise Gaussian process approximation can be written as $$f_0 = \mu_\vartheta + \Phi_\vartheta \xi_0,$$ where $\Phi_\vartheta \in \mathbb{R}^{m\times q}$ contains the basis or feature evaluations at $s_{1:m}$, and $\xi_0 \sim \mathcal{N}(0, I_q)$. For example, $$(\Phi_\vartheta)_{ik} = \sqrt{\lambda_{\vartheta,k}}\phi_{\vartheta,k}(s_i)$$ for a truncated Karhunen–Loève expansion. The innovation is given by $$\xi_0 \in \mathbb{R}^q, \qquad \xi_0 \sim \mathcal{N}(0, I_q).$$ The generator is given by $$T_\vartheta(\xi_0) = \mu_\vartheta + \Phi_\vartheta \xi_0.$$ The innovation dimension q need not equal the discretisation dimension m. When $q < m$, the induced prior on E_m is typically supported on a low-dimensional subspace unless an additional residual or nugget term is included. More elaborate pathwise Gaussian process constructions fit the same form after augmenting ξ_0 with auxiliary Gaussian variables.
Spectral random field	A finite spectral random field can be written as $$f(s) = \mu_\vartheta(s) + \sum_{k=1}^{q} A_{\vartheta,k}\big(a_k \cos(\omega_k^\top s) + b_k \sin(\omega_k^\top s)\big),$$ where $A_{\vartheta,k}$ are spectral amplitudes, ω_k are frequencies, and $a_k, b_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. Here the frequencies ω_k are treated as fixed, or as part of ϑ. If the frequencies are themselves random, they can be appended to the innovation variable. The innovation is given by $$\xi_0 = (a_1, b_1, \ldots, a_q, b_q) \in \mathbb{R}^{2q}, \qquad \xi_0 \sim \mathcal{N}(0, I_{2q}).$$ The generator is given by $$(T_\vartheta(\xi_0))_i = \mu_\vartheta(s_i) + \sum_{k=1}^{q} A_{\vartheta,k}\big(a_k \cos(\omega_k^\top s_i) + b_k \sin(\omega_k^\top s_i)\big),$$ for components $i = 1, \ldots, m$.
Gaussian Markov random field, or SPDE-defined Matérn field	A Gaussian Markov random field, including an SPDE-defined Matérn field after finite-element discretisation, can be written as $$R_\vartheta^\top R_\vartheta = Q_\vartheta, \qquad f_0 = \mu_\vartheta + A_\vartheta R_\vartheta^{-1}\xi_0,$$ where $x_\vartheta = R_\vartheta^{-1}\xi_0 \in \mathbb{R}^q$ is a vector of GMRF or finite-element coefficients, $Q_\vartheta \succ 0$ is its sparse precision matrix, $A_\vartheta : \mathbb{R}^q \to E_m$ maps the coefficients to the requested process values, and $\xi_0 \sim \mathcal{N}(0, I_q)$. For an SPDE-defined Matérn field, Q_ϑ is the sparse precision matrix of the finite-element coefficient vector, and A_ϑ is the corresponding finite-element evaluation or projection matrix. The innovation is given by $$\xi_0 \in \mathbb{R}^q, \qquad \xi_0 \sim \mathcal{N}(0, I_q).$$ The generator is given by $$T_\vartheta(\xi_0) = \mu_\vartheta + A_\vartheta R_\vartheta^{-1}\xi_0.$$ In practice, one solves a sparse triangular system rather than forming R_ϑ^{-1}. This includes proper areal Gaussian Markov random fields such as proper CAR and Leroux-type models, as well as SPDE-defined Matérn fields. Intrinsic CAR or intrinsic GMRF precision matrices are singular and instead require an identifying constraint, a reduced-coordinate factorisation on the constrained subspace, or an explicitly specified generalised-inverse construction.

continued on next page

Model class	Innovation representation
Intensity and convolution-field priors	
Log-Gaussian Cox intensity	A discretised log-Gaussian Cox process intensity can be written as $$f_0 = \exp(\mu_\vartheta + L_\vartheta \xi_0), \qquad L_\vartheta L_\vartheta^\top = K_\vartheta,$$ where $f_0 = (\lambda(s_1), \dots, \lambda(s_m))$, $g_0 = \mu_\vartheta + L_\vartheta \xi_0$ is the discretised latent Gaussian process, $\lambda(s_i) = \exp(g_0(s_i))$, and $\xi_0 \sim \mathcal{N}(0, I_m)$. The innovation is given by $$\xi_0 \in \mathbb{R}^m, \qquad \xi_0 \sim \mathcal{N}(0, I_m).$$ The generator is given by $$T_\vartheta(\xi_0) = \exp(\mu_\vartheta + L_\vartheta \xi_0),$$ where the exponential is applied componentwise.
Lévy-driven or convolution random field	A finite Lévy-driven or stable-convolution random field can be written as $$f(s_i) = \sum_{k=1}^{q} G_\vartheta(s_i - c_k) J_k, \qquad i = 1, \dots, m,$$ where G_ϑ is a convolution kernel, c_k are support or quadrature locations, and $J_k \sim \nu_{\vartheta,k}$ are increments of an independently scattered random measure. The natural innovation is given by $$\zeta_0 = (J_1, \dots, J_q).$$ Let $H_{\vartheta,k}$ be a map such that $$J_k = H_{\vartheta,k}(w_k), \qquad w_k \sim \mathcal{N}(0, I).$$ The Gaussian innovation is then given by $$\xi_0 = (w_1, \dots, w_q).$$ The corresponding generator is given by $$(T_\vartheta(\xi_0))_i = \sum_{k=1}^{q} G_\vartheta(s_i - c_k) H_{\vartheta,k}(w_k), \qquad i = 1, \dots, m.$$ This gives a non-Gaussian random-field prior, including Cauchy, normal-inverse Gaussian (NIG), stable, or other heavy-tailed convolution fields, through independently scattered random-measure increments.

continued on next page

Model class	Innovation representation
Discrete and extreme-value priors	
Potts or discrete Markov random field	A Potts or discrete Markov random field on sites $s_{1:m}$ can be written as $$\mathbb{P}_\vartheta(f_0 = x) \propto \exp\Big(\sum_{i=1}^{m} h_{\vartheta,i}(x_i) + \sum_{(i,j)\in\mathcal{E}} \beta_{\vartheta,ij}\mathbf{1}(x_i = x_j)\Big),$$ where $x = (x_1, \dots, x_m) \in \{1, \dots, K\}^m$, $x_i = f_0(s_i)$, $\mathcal{E}$ is the neighbourhood graph on $s_{1:m}$, $h_{\vartheta,i}$ are site potentials, and $\beta_{\vartheta,ij}$ are interaction parameters. The natural innovation can be represented by a simulator seed $$\zeta_0 = u \in (0,1)^q, \qquad u \sim \mathrm{Unif}((0,1)^q),$$ together with an exact sampler $\mathcal{S}_\vartheta$ such that $f_0 = \mathcal{S}_\vartheta(u) \sim \mathbb{P}_\vartheta$. If $\mathcal{S}_\vartheta$ is an approximate MCMC or sequential sampler, then the corresponding T_ϑ represents the algorithmic approximation rather than the exact Potts law. Let $\xi_0 = \Phi_{\mathrm{N}}^{-1}(u)$, where Φ_{N} is the standard normal distribution function, applied componentwise. The Gaussian innovation is then given by $$\xi_0 \in \mathbb{R}^q, \qquad \xi_0 \sim \mathcal{N}(0, I_q).$$ The corresponding generator is given by $$T_\vartheta(\xi_0) = \mathcal{S}_\vartheta(\Phi_{\mathrm{N}}(\xi_0)).$$ Exact Potts fields are discrete; gradient-based guidance therefore requires a relaxation, surrogate, or derivative-free guidance estimator.
Max-stable process	A finite spectral truncation of a simple max-stable process with unit Fréchet margins can be written as $$f(s_i) = \max_{1\le k\le K} P_k W_{\vartheta,k}(s_i), \qquad P_k = \Gamma_k^{-1}, \qquad \Gamma_k = \sum_{\ell=1}^{k} E_\ell,$$ where $E_k \overset{\text{i.i.d.}}{\sim} \mathrm{Exp}(1)$, the nonnegative spectral functions $W_{\vartheta,k}$ are independent and identically distributed, and the sequence $(W_{\vartheta,k})_{k\ge1}$ is independent of $(E_k)_{k\ge1}$. The spectral functions satisfy the usual normalisation $\mathbb{E}[W_{\vartheta,k}(s)] = 1$ for each s. For finite K, this is a spectral truncation of the corresponding infinite max-stable representation. The natural innovation is given by $$\zeta_0 = (E_{1:K}, \upsilon_{1:K}),$$ where υ_k generates the spectral function $W_{\vartheta,k}$. Let $$w_k^E = \Phi_{\mathrm{N}}^{-1}(\exp(-E_k)),$$ so that, equivalently, $$E_k = -\log\Phi_{\mathrm{N}}(w_k^E), \qquad w_k^E \sim \mathcal{N}(0,1).$$ If the spectral innovation can be Gaussianised as $$\upsilon_k = H_\vartheta^W(w_k^W), \qquad w_k^W \sim \mathcal{N}(0, I),$$ then the Gaussian innovation is given by $$\xi_0 = (w_1^E, \dots, w_K^E, w_1^W, \dots, w_K^W).$$ All variables w_k^E and w_k^W are taken to be mutually independent across k and across the two innovation blocks. The corresponding generator is given by $$(T_\vartheta(\xi_0))_i = \max_{1\le k\le K} \Gamma_k^{-1} W_\vartheta(s_i; H_\vartheta^W(w_k^W)), \qquad i = 1, \dots, m,$$ where $\Gamma_k = \sum_{\ell=1}^{k} -\log\Phi_{\mathrm{N}}(w_\ell^E)$. The maximum is non-smooth at ties, so smooth relaxations may be useful for gradient-based guidance.

continued on next page

Model class	Innovation representation
Dynamical and spatiotemporal priors	
State-space model or stochastic recurrence	A state-space model or stochastic recurrence can be written as $$x_0 = \Psi_{\vartheta,0}(\eta_0),$$ $$x_n = \Psi_{\vartheta,n}(x_{n-1}, \eta_n), \qquad n = 1, \dots, N,$$ where $\eta_n \overset{\text{ind}}{\sim} r_n$ are model innovations and $\Psi_{\vartheta,n}$ is the one-step simulator. The natural innovation is given by $$\zeta_0 = (\eta_0, \dots, \eta_N).$$ Let $H_{\vartheta,n}$ be a map such that $$\eta_n = H_{\vartheta,n}(w_n), \qquad w_n \sim \mathcal{N}(0, I).$$ The Gaussian innovation is then given by $$\xi_0 = (w_0, \dots, w_N).$$ The corresponding generator is given by $$x_0 = \Psi_{\vartheta,0}(H_{\vartheta,0}(w_0)),$$ $$x_n = \Psi_{\vartheta,n}(x_{n-1}, H_{\vartheta,n}(w_n)), \qquad n = 1, \dots, N,$$ with output $$T_\vartheta(\xi_0) = (x_0, \dots, x_N).$$ When the original innovations are already Gaussian, $H_{\vartheta,n}$ is the identity map. For discrete or mixed innovations, the maps $H_{\vartheta,n}$ may be nonsmooth, in which case gradient-based guidance again requires a relaxation, surrogate, or derivative-free estimator. Linear Gaussian and nonlinear Gaussian state-space models correspond to particular choices of $\Psi_{\vartheta,n}$.
Diffusion process	A diffusion process satisfying $$\mathrm{d}x_\tau = b_\vartheta(\tau, x_\tau)\,\mathrm{d}\tau + \Sigma_\vartheta(\tau, x_\tau)\,\mathrm{d}w_\tau$$ can be discretised by Euler–Maruyama as $$x_{n+1} = x_n + b_\vartheta(\tau_n, x_n)\Delta_n + \Sigma_\vartheta(\tau_n, x_n)\sqrt{\Delta_n}\eta_n,$$ where $\Delta_n = \tau_{n+1} - \tau_n$, $\eta_n \overset{\text{ind}}{\sim} \mathcal{N}(0, I)$, and $\Sigma_\vartheta(\tau_n, x_n)$ maps Gaussian innovations into diffusion increments. The innovation is given by $$\xi_0 = (\eta_{-1}, \eta_0, \dots, \eta_{N-1}),$$ where $\eta_{-1} \sim \mathcal{N}(0, I)$ generates the initial state if x_0 is random, through an initial-state map $x_0 = \chi_{\vartheta,0}(\eta_{-1})$, and is omitted otherwise. The generator is given by applying the Euler–Maruyama recursion above and returning $$T_\vartheta(\xi_0) = (x_0, \dots, x_N).$$ More generally, higher-order numerical solvers correspond to different choices of T_ϑ.

continued on next page

Model class	Innovation representation
Jump diffusion, Lévy-driven SDE, or Lévy process	A jump diffusion or Lévy-driven SDE satisfying $$\mathrm{d}x_\tau = b_\vartheta(\tau, x_\tau)\,\mathrm{d}\tau + \Sigma_\vartheta(\tau, x_\tau)\,\mathrm{d}W_\tau + \mathrm{d}j_\tau$$ can be discretised as $$x_{n+1} = x_n + b_\vartheta(\tau_n, x_n)\Delta_n + \Sigma_\vartheta(\tau_n, x_n)\sqrt{\Delta_n}\eta_n + \ell_n,$$ where $\Delta_n = \tau_{n+1} - \tau_n$, $\eta_n \overset{\text{ind}}{\sim} \mathcal{N}(0, I)$, and $\ell_n \sim \nu_{\vartheta,\Delta_n}$ is the jump or Lévy increment over $[\tau_n, \tau_{n+1}]$. The natural innovation is given by $$\zeta_0 = (\eta_{-1}, \eta_0, \ldots, \eta_{N-1}, \ell_0, \ldots, \ell_{N-1}),$$ where $\eta_{-1} \sim \mathcal{N}(0, I)$ generates the initial state if x_0 is random, through an initial-state map $x_0 = \chi_{\vartheta,0}(\eta_{-1})$, and is omitted otherwise. Let H_{ϑ,Δ_n} be a map such that $$\ell_n = H_{\vartheta,\Delta_n}(w_n), \qquad w_n \sim \mathcal{N}(0, I).$$ The Gaussian innovation is then given by $$\xi_0 = (\eta_{-1}, \eta_0, \ldots, \eta_{N-1}, w_0, \ldots, w_{N-1}).$$ The corresponding generator is given by applying the jump–Euler recursion with $$\ell_n = H_{\vartheta,\Delta_n}(w_n)$$ and returning $$T_\vartheta(\xi_0) = (x_0, \ldots, x_N).$$ A Lévy process or subordinator is obtained as the special case with no state-dependent drift or diffusion term and additive Lévy increments.
Time-discretised SPDE	A spatially discretised SPDE of the form $$\mathrm{d}u_\tau = \mathcal{A}_\vartheta(u_\tau)\,\mathrm{d}\tau + \mathcal{B}_\vartheta(u_\tau)\,\mathrm{d}W_\tau$$ can be time-discretised as $$u_{n+1} = \Psi_{\vartheta,n}(u_n, \eta_n), \qquad n = 0, \ldots, N-1,$$ where $\eta_n \overset{\text{ind}}{\sim} \mathcal{N}(0, I)$ are finite-dimensional Gaussian noise innovations and $\Psi_{\vartheta,n}$ is the numerical SPDE time-stepper. The innovation is given by $$\xi_0 = (\eta_{-1}, \eta_0, \ldots, \eta_{N-1}),$$ where $\eta_{-1} \sim \mathcal{N}(0, I)$ generates the initial state if u_0 is random, through an initial-state map $u_0 = \chi_{\vartheta,0}(\eta_{-1})$, and is omitted otherwise. The generator is given by applying the numerical SPDE solver above and returning the discretised space-time field.

continued on next page

Model class	Innovation representation
Implicit and simulator-defined priors	
Neural or deep implicit process	A neural or deep implicit process can be written as $$f(s_i) = G_\vartheta(s_i, h, \epsilon_i), \qquad i = 1, \dots, m,$$ where G_ϑ is a neural decoder, h is a global latent variable, and ϵ_i are optional local noise variables. The innovation is given by $$\xi_0 = (h, \epsilon_1, \dots, \epsilon_m),$$ with a simple reference law, typically standard Gaussian. The generator is given by $$(T_\vartheta(\xi_0))_i = G_\vartheta(s_i, h, \epsilon_i), \qquad i = 1, \dots, m.$$ If the decoder is differentiable, $D_\xi T_\vartheta$ is available by automatic differentiation.
Black-box simulator	A black-box simulator can be written as $$f_0 = \mathcal{S}_\vartheta(s_{1:m}; u),$$ where $\mathcal{S}_\vartheta$ is the simulator and $u \sim \mathrm{Unif}((0,1)^q)$ is the simulator random seed. The natural innovation is given by $$\zeta_0 = u \in (0,1)^q.$$ Let $\xi_0 = \Phi_{\mathrm{N}}^{-1}(u)$, where Φ_{N} is the standard normal distribution function, applied componentwise. Then $$\xi_0 \sim \mathcal{N}(0, I_q).$$ The corresponding generator is given by $$T_\vartheta(\xi_0) = \mathcal{S}_\vartheta(s_{1:m}; \Phi_{\mathrm{N}}(\xi_0)).$$ If $\mathcal{S}_\vartheta$ is not differentiable, guidance requires a smooth surrogate, relaxation, or derivative-free estimator.

B Proofs for the latent conditioning principle

Proof of Prop. 2.2. Let $\xi_0 \sim \rho_\vartheta(\cdot \mid \mathcal{C})$. Then, for any measurable $A \subseteq E_m$, we have that

$$\begin{aligned}\mathbb{P}\big(T_\vartheta(\xi_0) \in A\big) &= \frac{1}{Z_\vartheta} \int_{\mathbb{R}^{m_\xi}} \mathbb{1}_{\{T_\vartheta(\xi_0) \in A\}} L_{\mathcal{C}}(T_\vartheta(\xi_0))\, r(\xi_0)\, \mathrm{d}\xi_0 \\ &= \frac{1}{Z_\vartheta} \int_A L_{\mathcal{C}}(f_0)\, p_\vartheta(f_0)\, \mathrm{d}f_0 = \int_A \pi_\vartheta(f_0 \mid \mathcal{C})\, \mathrm{d}f_0.\end{aligned}$$

The third equality follows from the pushforward relation $T_{\vartheta\#}(r(\xi_0)\,\mathrm{d}\xi_0) = p_\vartheta(f_0)\,\mathrm{d}f_0$, applied to the nonnegative measurable function $g(f_0) = \mathbf{1}_A(f_0) L_{\mathcal{C}}(f_0)$. Hence $T_\vartheta(\xi_0)$ has density $\pi_\vartheta(\cdot \mid \mathcal{C})$, as required. □

Remark B.1. We use density notation in the main text to simplify the exposition. Our approach, however, admits a much more general measure-theoretic formulation. Let P_ϑ denote the prior law of f_0. Assume that $L_{\mathcal{C}} : E_m \to [0, \infty)$ is measurable. The conditional law on E_m is then

$$\Pi_\vartheta(df_0 \mid \mathcal{C}) = \frac{L_{\mathcal{C}}(f_0) P_\vartheta(df_0)}{Z_\vartheta}, \qquad Z_\vartheta := \int L_{\mathcal{C}}(f_0) P_\vartheta(df_0). \tag{5}$$

Our assumption now reads as follows. There exist a latent dimension m_ξ, a reference probability measure R on $\mathbb{R}^{m_\xi}$, and a measurable map $T_\vartheta : \mathbb{R}^{m_\xi} \to E_m$ such that $P_\vartheta = (T_\vartheta)_{\#} R$. Equivalently, if $\xi_0 \sim R$, then $f_0 = T_\vartheta(\xi_0) \sim P_\vartheta$. The corresponding latent target is given by

$$\rho_\vartheta(d\xi \mid \mathcal{C}) = \frac{L_{\mathcal{C}}(T_\vartheta(\xi)) R(d\xi)}{Z_\vartheta}, \qquad Z_\vartheta = \int L_{\mathcal{C}}(T_\vartheta(\xi)) R(d\xi). \tag{6}$$

We emphasise that the values of Z_ϑ as defined in (5) and (6) coincide, since $P_\vartheta = T_{\vartheta\#} R$ implies $\int L_{\mathcal{C}}(f_0) P_\vartheta(df_0) = \int L_{\mathcal{C}}(T_\vartheta(\xi)) R(d\xi)$ by definition of the pushforward. Now, for any

measurable set $A \subseteq E_m$, we have that

$$T_{\vartheta\#}\rho_\vartheta(\cdot \mid \mathcal{C})(A) = \int \mathbb{1}_{\{T_\vartheta(\xi)\in A\}} \frac{L_{\mathcal{C}}(T_\vartheta(\xi))R(d\xi)}{Z_\vartheta} = \frac{1}{Z_\vartheta}\int_A L_{\mathcal{C}}(f_0)P_\vartheta(df_0) = \Pi_\vartheta(A \mid \mathcal{C}).$$

It follows that $T_{\vartheta\#}\rho_\vartheta(\cdot \mid \mathcal{C}) = \Pi_\vartheta(\cdot \mid \mathcal{C})$. This is the measure-theoretic form of Prop. 2.2. The density-based statement in the main text is recovered when P_ϑ admits a density p_ϑ. This more general argument also covers singular and discrete laws.

Our next result establishes that the latent representation also preserves the relative-entropy cost of conditioning.

Proposition B.2. *Suppose that Assumption 2.1 holds and that $0 < Z_\vartheta < \infty$. Then, in the extended sense,*

$$\mathrm{KL}\big(\rho_\vartheta(\cdot \mid \mathcal{C}) \,\big\|\, r\big) = \mathrm{KL}\big(\pi_\vartheta(\cdot \mid \mathcal{C}) \,\big\|\, p_\vartheta\big)\,.$$

Proof. By construction, we have

$$\frac{\rho_\vartheta(\xi_0 \mid \mathcal{C})}{r(\xi_0)} = \frac{\Lambda_{\vartheta,\mathcal{C}}(\xi_0)}{Z_\vartheta} = \frac{L_{\mathcal{C}}(T_\vartheta(\xi_0))}{Z_\vartheta} \qquad r\text{-a.e.} \tag{7}$$

Similarly, due to (1), it holds that

$$\frac{\pi_\vartheta(f_0 \mid \mathcal{C})}{p_\vartheta(f_0)} = \frac{L_{\mathcal{C}}(f_0)}{Z_\vartheta} \qquad p_\vartheta\text{-a.e.} \tag{8}$$

Let $\ell(f_0) = \log\left(\frac{L_{\mathcal{C}}(f_0)}{Z_\vartheta}\right)$, with the usual extended-value convention. Using (7), we obtain

$$\mathrm{KL}\big(\rho_\vartheta(\cdot \mid \mathcal{C}) \,\big\|\, r\big) = \int_{\mathbb{R}^{m_\xi}} \ell(T_\vartheta(\xi_0))\rho_\vartheta(\xi_0 \mid \mathcal{C})\,\mathrm{d}\xi_0.$$

By Prop. 2.2, pushing $\rho_\vartheta(\cdot \mid \mathcal{C})$ forward through T_ϑ gives $\pi_\vartheta(\cdot \mid \mathcal{C})$. We therefore have,

$$\int_{\mathbb{R}^{m_\xi}} \ell(T_\vartheta(\xi_0))\rho_\vartheta(\xi_0 \mid \mathcal{C})\,\mathrm{d}\xi_0 = \int_{E_m} \ell(f_0)\pi_\vartheta(f_0 \mid \mathcal{C})\,\mathrm{d}f_0. \tag{9}$$

Finally, using (8), the right-hand side of (9) is exactly $\mathrm{KL}\big(\pi_\vartheta(\cdot \mid \mathcal{C}) \,\big\|\, p_\vartheta\big)$. This proves the identity. □

This result suggests that the latent construction is not just a convenient sampling device. Since the additional likelihood depends on the latent variable only through the generated process value $T_\vartheta(\xi_0)$, the relative-entropy deformation from the prior p_ϑ to the conditional $\pi_\vartheta(\cdot \mid \mathcal{C})$ is exactly matched by the deformation from the reference r to the latent conditional $\rho_\vartheta(\cdot \mid \mathcal{C})$. In contrast, for a generic latent density η on $\mathbb{R}^{m_\xi}$, if p_η denotes the density of $T_\vartheta(\xi_0)$ when $\xi_0 \sim \eta$, then the data-processing inequality only gives $\mathrm{KL}(p_\eta \,\|\, p_\vartheta) \le \mathrm{KL}(\eta \,\|\, r)$. Thus the latent target introduces no additional information beyond the output-level conditioning likelihood.

C Exactness of the latent dynamics

We now fix ϑ and $\mathcal{C}$, and write $T = T_\vartheta$, $\Lambda = \Lambda_{\vartheta,\mathcal{C}}$, $Z = Z_\vartheta$, and $\rho_0(\xi_0) = \rho_\vartheta(\xi_0 \mid \mathcal{C})$. We also write $r(\xi) = \mathcal{N}(\xi; 0, I_{m_\xi})$.

Recall that the VP noising schedule is specified by a locally integrable function $\beta : [0,1) \to (0,\infty)$. We write

$$B(t) = \int_0^t \beta(s)\,\mathrm{d}s, \qquad \alpha(t) = \exp\big\{-\frac{1}{2}B(t)\big\}, \qquad \sigma(t)^2 = 1 - \alpha(t)^2.$$

We now allow either a finite-noising horizon, where $B(1) := \lim_{t\uparrow 1} B(t) < \infty$, or an ideal infinite-noising horizon, where $B(1) = \infty$ and hence $\alpha(t) \downarrow 0$ as $t \uparrow 1$. For $t \in [0, 1)$, the VP bridge is

$$\xi_t = \alpha(t)\xi_0 + \sigma(t)\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_{m_\xi}), \tag{10}$$

where ε is independent of ξ_0. We write ρ_t for the density of the noised latent variable obtained by drawing $\xi_0 \sim \rho_0$ and applying (10). Equivalently,

$$\rho_t(u) = \frac{\psi(t,u)r(u)}{Z}, \qquad \psi(t,u) = \mathbb{E}_{\xi_0\sim r}\left[\Lambda(\xi_0) \mid \xi_t = u\right].$$

where the conditional expectation is taken under the reference bridge with $\xi_0 \sim r$. Finally, assuming $\psi(t,\cdot)$ is differentiable, we write $g(t,u) = \nabla_u \log \psi(t,u)$ for the latent guidance field.

In Sec. 3, we formally identified the SDE associated with the guided latent marginals $(\rho_t)_{0\le t<1}$. We now recall these dynamics using notation that is convenient for the exactness statements below. The forward noising SDE is given by

$$\mathrm{d}\xi_t = -\frac{1}{2}\beta(t)\xi_t \,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}w_t, \qquad \xi_0 \sim \rho_0, \qquad 0 \le t < 1,$$

where $w = (w_t)_{0\le t<1}$ is a standard $\mathbb{R}^{m_\xi}$-valued Brownian motion. Fix $0 \le t_0 < t_1 < 1$, and set $\tau_s = t_1 - s$, for $0 \le s \le t_1 - t_0$. The reverse-time SDE is then given by

$$\mathrm{d}\xi^{\leftarrow}_s = \left[\beta(\tau_s)g(\tau_s, \xi^{\leftarrow}_s) - \frac{1}{2}\beta(\tau_s)\xi^{\leftarrow}_s\right]\mathrm{d}s + \sqrt{\beta(\tau_s)}\,\mathrm{d}\overline{w}_s, \qquad \xi^{\leftarrow}_0 \sim \rho_{t_1}, \tag{11}$$

where $\overline{w} = (\overline{w}_s)_{0\le s\le t_1-t_0}$ is a standard Brownian motion in reverse time. The same family of marginals can also be generated by a deterministic process known as the probability-flow ODE (e.g., Song et al., 2021). In forward noising time, this ODE is given by

$$\frac{\mathrm{d}u_t}{\mathrm{d}t} = -\frac{1}{2}\beta(t)g(t,u_t), \qquad u_0 \sim \rho_0, \qquad 0 \le t < 1. \tag{12}$$

For sampling, we again use the reverse-time parametrisation $u^{\leftarrow}_s = u_{\tau_s}$, with τ_s as defined above. In particular, the reverse-time probability-flow ODE is

$$\frac{\mathrm{d}u^{\leftarrow}_s}{\mathrm{d}s} = \frac{1}{2}\beta(\tau_s)g(\tau_s, u^{\leftarrow}_s), \qquad u^{\leftarrow}_0 \sim \rho_{t_1}. \tag{13}$$

Assumption C.1. For every compact interval $[t_0, t_1] \subset (0,1)$, ρ_t is a strictly positive Lebesgue density for all $t \in [t_0, t_1]$. The map $(t,u) \mapsto \nabla_u \log \rho_t(u)$ is sufficiently regular that the reverse-time SDE (11) and the probability-flow ODE (13) are well posed on the corresponding time intervals, and their associated Fokker–Planck and continuity equations have unique density-valued solutions.

Theorem C.2. *Suppose that Assumption 2.1 holds, that $r = \mathcal{N}(0, I_{m_\xi})$, that $0 < Z < \infty$, and that Assumption C.1 holds. Fix $0 < t_0 < t_1 < 1$, and define $\tau_s = t_1 - s$ for $0 \le s \le t_1 - t_0$. Let $\xi^{\leftarrow}_0 \sim \rho_{t_1}$, and let $(\xi^{\leftarrow}_s)_{0\le s\le t_1-t_0}$ solve the reverse-time SDE (11). Then $\xi^{\leftarrow}_s \sim \rho_{\tau_s}$ for all $0 \le s \le t_1 - t_0$. In particular, $\xi^{\leftarrow}_{t_1-t_0} \sim \rho_{t_0}$. If the reverse SDE is well defined down to $t_0 = 0$, then*

$$\xi^{\leftarrow}_{t_1} \sim \rho_0 = \rho_\vartheta(\cdot \mid \mathcal{C}), \qquad T_\vartheta(\xi^{\leftarrow}_{t_1}) \sim \pi_\vartheta(\cdot \mid \mathcal{C}).$$

Proof. The proof is standard. Let $b_t(u) = -\frac{1}{2}\beta(t)u$ denote the drift of the forward VP SDE. By the standard time-reversal formula for diffusions (e.g., Anderson, 1982), the reverse-time diffusion with clock $\tau_s = t_1 - s$ has drift given by

$$\widetilde{b}_s(u) = -b_{\tau_s}(u) + \beta(\tau_s)\nabla_u \log \rho_{\tau_s}(u). \tag{14}$$

Moreover, since $\rho_t(u) = Z^{-1}\psi(t,u)r(u)$ and $r(u) = \mathcal{N}(u; 0, I_{m_\xi})$, it follows straightforwardly that

$$\nabla_u \log \rho_t(u) = \nabla_u \log \psi(t,u) - u = g(t,u) - u. \tag{15}$$

We thus have, substituting (15) into (14), that

$$\widetilde{b}_s(u) = \frac{1}{2}\beta(\tau_s)u + \beta(\tau_s)\left(g(\tau_s,u) - u\right) = \beta(\tau_s)g(\tau_s,u) - \frac{1}{2}\beta(\tau_s)u.$$

This is exactly the drift in (11). Equivalently, if $q_s(u) = \rho_{\tau_s}(u)$, then q_s solves the Fokker–Planck equation associated with (11). Indeed,

$$\partial_s q_s = -\partial_t \rho_t\big|_{t=\tau_s} = \nabla \cdot (b_{\tau_s} \rho_{\tau_s}) - \frac{1}{2}\beta(\tau_s)\Delta\rho_{\tau_s} = -\nabla \cdot (\widetilde{b}_s q_s) + \frac{1}{2}\beta(\tau_s)\Delta q_s.$$

By uniqueness in Assumption C.1, the law of $\xi_s^{\leftarrow}$ has density $q_s = \rho_{\tau_s}$. The claims at $t_0 = 0$ follow because $\rho_0 = \rho_\vartheta(\cdot \mid \mathcal{C})$. Finally, the process-space statement follows from Prop. 2.2. □

Theorem C.3. *Suppose that Assumption 2.1 holds, that $r = \mathcal{N}(0, I_{m_\xi})$, that $0 < Z < \infty$, and that Assumption C.1 holds. Fix $0 < t_0 < t_1 < 1$, and define $\tau_s = t_1 - s$ for $0 \le s \le t_1 - t_0$. If $u_0^{\leftarrow} \sim \rho_{t_1}$ and $(u_s^{\leftarrow})_{0 \le s \le t_1 - t_0}$ solves the reverse-time probability-flow ODE (13), then $u_s^{\leftarrow} \sim \rho_{\tau_s}$ for all $0 \le s \le t_1 - t_0$. In particular, $u_{t_1 - t_0}^{\leftarrow} \sim \rho_{t_0}$. If the flow is well defined down to $t_0 = 0$, then*

$$u_{t_1}^{\leftarrow} \sim \rho_0 = \rho_\vartheta(\cdot \mid \mathcal{C}), \qquad T_\vartheta(u_{t_1}^{\leftarrow}) \sim \pi_\vartheta(\cdot \mid \mathcal{C}).$$

Proof. The marginals $(\rho_t)_{0 \le t < 1}$ are generated by the forward VP-SDE when the initial law is $\rho_0 = \rho_\vartheta(\cdot \mid \mathcal{C})$. Indeed, for any bounded measurable test function φ,

$$\begin{aligned}\mathbb{E}_{\xi_0 \sim \rho_0}[\varphi(\xi_t)] &= \frac{1}{Z}\mathbb{E}_{\xi_0 \sim r, \varepsilon \sim r}\left[\Lambda(\xi_0)\varphi\left(\alpha(t)\xi_0 + \sigma(t)\varepsilon\right)\right] \\ &= \frac{1}{Z}\mathbb{E}_{u \sim r}\left[\varphi(u)\mathbb{E}\left[\Lambda(\xi_0) \mid \xi_t = u\right]\right] = \int_{\mathbb{R}^{m_\xi}} \varphi(u)\rho_t(u)\,\mathrm{d}u.\end{aligned}$$

Thus ρ_t solves the Fokker–Planck equation

$$\partial_t \rho_t = -\nabla \cdot (b_t \rho_t) + \frac{1}{2}\beta(t)\Delta\rho_t, \qquad b_t(u) = -\frac{1}{2}\beta(t)u. \tag{16}$$

Writing $s_t(u) = \nabla_u \log \rho_t(u)$, we can rewrite (16) as

$$\partial_t \rho_t = -\nabla \cdot (b_t \rho_t) + \frac{1}{2}\beta(t)\nabla \cdot (\rho_t s_t) = -\nabla \cdot \big(\big[b_t - \frac{1}{2}\beta(t)s_t\big]\rho_t\big).$$

It follows that ρ_t also solves the continuity equation

$$\partial_t \rho_t + \nabla \cdot (v_t \rho_t) = 0, \qquad v_t(u) = b_t(u) - \frac{1}{2}\beta(t)s_t(u).$$

Meanwhile, substituting (15), we have that

$$v_t(u) = -\frac{1}{2}\beta(t)u - \frac{1}{2}\beta(t)\left(g(t,u) - u\right) = -\frac{1}{2}\beta(t)g(t,u).$$

This is exactly the vector field in the forward probability-flow ODE (12). By Assumption C.1, the ODE flow is well posed and gives the unique solution of the continuity equation. Therefore the backward flow (13) transports ρ_{t_1} to ρ_{t_0}. Taking $t_0 = 0$ gives $u_{t_1}^{\leftarrow} \sim \rho_0 = \rho_\vartheta(\cdot \mid \mathcal{C})$, and the process-space statement follows from Prop. 2.2. □

Remark C.4. If $B(1) = \infty$, then $\alpha(t) \downarrow 0$ and the noised target marginals satisfy $\rho_t \to r$ in total variation as $t \uparrow 1$, provided $\Lambda \in L^1(r)$. In this sense, starting from $u_0^{\leftarrow} \sim r$ can be viewed as the limiting infinite-noising version of Theorem C.3. For a finite noising horizon, exact sampling from a terminal time t_1 requires initialising from the terminal marginal ρ_{t_1}, not from r. If one instead starts from r, the initialisation discrepancy is $\|\rho_{t_1} - r\|_{\mathrm{TV}}$, and the discrepancy after exact reverse probability flow is at most this quantity, with equality when the deterministic flow is invertible.

D Approximation and stability guarantees

The practical sampler replaces the exact guidance field $g(t,u)$ by an approximate field $\widehat{g}_S(t,u)$, and then solves the resulting ODE or SDE numerically. This introduces two sources of error: the guidance approximation error and the numerical discretisation error. Both errors enter the probability-flow ODE and the reverse-time SDE through the same perturbation mechanism. We thus state the main stability and numerical-error bounds in a unified form.

D.1 Common perturbation framework

We work on a truncated forward-time interval $[\tau, t_1]$, where $0 < \tau < t_1 < 1$, and use reverse-time notation throughout this subsection. Set

$$R_{\tau,t_1} = t_1 - \tau, \qquad \kappa_s = t_1 - s, \qquad 0 \le s \le R_{\tau,t_1}.$$

Thus s denotes reverse time and κ_s denotes the corresponding forward noising time. We index the two reverse dynamics by $\chi \in \{\text{ode}, \text{sde}\}$, and define

χ	λ_χ	ρ_χ	$\Sigma_\chi(s)$
ode	$\frac{1}{2}$	0	0
sde	1	$\frac{1}{2}$	$\sqrt{\beta(\kappa_s)} I_{m_\xi}$.

For an exact guidance field g and an approximate guidance field $\widehat{g}$, define the exact and approximate reverse-time drifts

$$F_\chi(s,u) = \lambda_\chi \beta(\kappa_s) g(\kappa_s, u) - \rho_\chi \beta(\kappa_s) u, \qquad \widehat{F}_\chi(s,u) = \lambda_\chi \beta(\kappa_s) \widehat{g}(\kappa_s, u) - \rho_\chi \beta(\kappa_s) u. \quad (17)$$

In particular, for the probability-flow ODE we have

$$a(s,u) = F_{\text{ode}}(s,u) = \frac{1}{2}\beta(\kappa_s) g(\kappa_s, u),$$
$$\widehat{a}(s,u) = \widehat{F}_{\text{ode}}(s,u) = \frac{1}{2}\beta(\kappa_s) \widehat{g}(\kappa_s, u).$$

Meanwhile, for the reverse-time SDE we have

$$b(s,u) = F_{\text{sde}}(s,u) = \beta(\kappa_s) g(\kappa_s, u) - \frac{1}{2}\beta(\kappa_s) u,$$
$$\widehat{b}(s,u) = \widehat{F}_{\text{sde}}(s,u) = \beta(\kappa_s) \widehat{g}(\kappa_s, u) - \frac{1}{2}\beta(\kappa_s) u.$$

The exact and approximate dynamics may then be written as

$$\mathrm{d}X_s^\chi = F_\chi(s, X_s^\chi)\,\mathrm{d}s + \Sigma_\chi(s)\,\mathrm{d}W_s, \qquad \mathrm{d}\widehat{X}_s^\chi = \widehat{F}_\chi(s, \widehat{X}_s^\chi)\,\mathrm{d}s + \Sigma_\chi(s)\,\mathrm{d}W_s. \quad (18)$$

When $\chi = \text{ode}$, we write $X^{\text{ode}} = u^{\leftarrow}, \widehat{X}^{\text{ode}} = \widehat{u}^{\leftarrow}$. When $\chi = \text{sde}$, we write $X^{\text{sde}} = \xi^{\leftarrow}, \widehat{X}^{\text{sde}} = \widehat{\xi}^{\leftarrow}$.

Let $\mathcal{R}_{\tau,t_1} \subset [\tau, t_1] \times \mathbb{R}^{m_\xi}$ be a region on which the exact and approximate guidance fields are compared, and which contains the exact and approximate trajectories under consideration. Define the realised pointwise guidance error by

$$\varepsilon(s) = \sup\Big\{ \|g(\kappa_s, u) - \widehat{g}(\kappa_s, u)\| : (\kappa_s, u) \in \mathcal{R}_{\tau,t_1} \Big\}, \qquad \delta_\chi(s) = \lambda_\chi \beta(\kappa_s) \varepsilon(s). \quad (19)$$

For the ODE and the SDE, we thus have that

$$\delta_{\text{ode}}(s) = \frac{1}{2}\beta(\kappa_s)\varepsilon(s), \qquad \delta_{\text{sde}}(s) = \beta(\kappa_s)\varepsilon(s).$$

In the common case where $\varepsilon_{\tau,t_1} := \sup_{(t,u)\in\mathcal{R}_{\tau,t_1}} \|g(t,u) - \widehat{g}(t,u)\| < \infty$, it follows from (19) that $\delta_\chi(s) \le \lambda_\chi \beta(\kappa_s) \varepsilon_{\tau,t_1}$.

Given a time-dependent one-sided Lipschitz rate $\eta \in L^1([0, R_{\tau,t_1}])$, we define the integral

$$\Theta_\eta(s,r) = \exp\left\{ \int_r^s \eta(q)\,\mathrm{d}q \right\}, \qquad 0 \le r \le s \le R_{\tau,t_1}.$$

Finally, throughout the statements below, we treat $\widehat{g}$ as a realised approximate guidance field. If $\widehat{g}$ is random, our results are to be read conditionally on this realisation. Probabilistic rates follow by combining these conditional stability bounds with separate probabilistic control of the realised guidance error.

D.2 Unified stability of approximate guidance

The following result is a variable-rate version of the deterministic perturbation bound obtained in (Moss et al., 2026, Proposition E.5), restated in our present notation and extended to cover the reverse-time SDE under a synchronous coupling.

Proposition D.1. *Fix $\chi \in \{\mathrm{ode}, \mathrm{sde}\}$, and let X^χ and $\widehat{X}^\chi$ solve (18) on $[0, R_{\tau,t_1}]$. For the SDE, assume that the two equations are driven by the same Brownian motion. Assume that $(\kappa_s, X_s^\chi), (\kappa_s, \widehat{X}_s^\chi) \in \mathcal{R}_{\tau,t_1}$ for all $s \in [0, R_{\tau,t_1}]$. Suppose that the exact reverse-time drift satisfies the one-sided Lipschitz condition*

$$\langle x - y, F_\chi(s,x) - F_\chi(s,y)\rangle \le \eta_\chi(s)\|x-y\|^2 \tag{20}$$

whenever (κ_s, x) and (κ_s, y) belong to $\mathcal{R}_{\tau,t_1}$. Then, for every $0 \le s \le R_{\tau,t_1}$, the following bound holds:

$$\|X_s^\chi - \widehat{X}_s^\chi\| \le \Theta_{\eta_\chi}(s,0)\|X_0^\chi - \widehat{X}_0^\chi\| + \int_0^s \Theta_{\eta_\chi}(s,r)\delta_\chi(r)\,\mathrm{d}r. \tag{21}$$

If the exact and approximate solves are coupled by the same reverse-time initial point, equivalently by the same terminal point at forward time t_1, the first term is absent. Moreover, for $p \ge 1$, provided the initial laws have finite p-th moments,

$$W_p\Big(\mathcal{L}(X_s^\chi), \mathcal{L}(\widehat{X}_s^\chi)\Big) \le \Theta_{\eta_\chi}(s,0) W_p\Big(\mathcal{L}(X_0^\chi), \mathcal{L}(\widehat{X}_0^\chi)\Big) + \int_0^s \Theta_{\eta_\chi}(s,r)\delta_\chi(r)\,\mathrm{d}r.$$

Proof. Let $D_s = X_s^\chi - \widehat{X}_s^\chi$. When $\chi = \mathrm{ode}$, this is an ordinary deterministic difference equation. When $\chi = \mathrm{sde}$, the Brownian terms cancel under the synchronous coupling. In either case, we have

$$D_s = D_0 + \int_0^s \{F_\chi(r, X_r^\chi) - \widehat{F}_\chi(r, \widehat{X}_r^\chi)\}\,\mathrm{d}r.$$

We can write the integrand as $F_\chi(r, X_r^\chi) - \widehat{F}_\chi(r, \widehat{X}_r^\chi) = F_\chi(r, X_r^\chi) - F_\chi(r, \widehat{X}_r^\chi) + F_\chi(r, \widehat{X}_r^\chi) - \widehat{F}_\chi(r, \widehat{X}_r^\chi)$. The first term is controlled by (20). The second term satisfies

$$\|F_\chi(r,u) - \widehat{F}_\chi(r,u)\| = \lambda_\chi \beta(\kappa_r)\|g(\kappa_r,u) - \widehat{g}(\kappa_r,u)\| \le \delta_\chi(r)$$

on the region under consideration. Hence the upper right-hand derivative satisfies

$$\frac{\mathrm{d}^+}{\mathrm{d}s}\|D_s\| \le \eta_\chi(s)\|D_s\| + \delta_\chi(s).$$

The variable-coefficient Gronwall inequality gives (21). The Wasserstein estimate follows by coupling the initial variables optimally and, in the SDE case, using the same Brownian motion for the two SDEs. □

Corollary D.2. *Suppose the conditions of Prop. D.1 hold. Suppose in addition that $\eta_\chi(s) \le \bar{\eta}_{\chi,\tau,t_1}$ on $[0, R_{\tau,t_1}]$ and $\bar{\beta}_{\tau,t_1} = \operatorname{ess\,sup}_{t\in[\tau,t_1]} \beta(t) < \infty$. Then, under common terminal coupling, it holds for all $s \in [0, R_{\tau,t_1}]$ that*

$$\|X_s^\chi - \widehat{X}_s^\chi\| \le \lambda_\chi \bar{\beta}_{\tau,t_1}\varepsilon_{\tau,t_1}\Psi_{\bar{\eta}_{\chi,\tau,t_1}}(s), \qquad \Psi_\eta(r) = \begin{cases} \frac{e^{\eta r}-1}{\eta}, & \eta \ne 0, \\ r, & \eta = 0. \end{cases}$$

Proof. Under common terminal coupling, the first term in (21) vanishes. Moreover, since $\eta_\chi(s) \le \bar{\eta}_{\chi,\tau,t_1}$ and $\delta_\chi(s) \le \lambda_\chi \bar{\beta}_{\tau,t_1}\varepsilon_{\tau,t_1}$, we have $\Theta_{\eta_\chi}(s,r) \le \exp\{\bar{\eta}_{\chi,\tau,t_1}(s-r)\}$. Therefore

$$\|X_s^\chi - \widehat{X}_s^\chi\| \le \lambda_\chi \bar{\beta}_{\tau,t_1}\varepsilon_{\tau,t_1}\int_0^s \exp\{\bar{\eta}_{\chi,\tau,t_1}(s-r)\}\,\mathrm{d}r = \lambda_\chi \bar{\beta}_{\tau,t_1}\varepsilon_{\tau,t_1}\Psi_{\bar{\eta}_{\chi,\tau,t_1}}(s).$$

□

D.3 Curvature regimes

The sign of the one-sided Lipschitz rate η_χ determines the nature of the bound in Prop. D.1 and Corollary D.2. This, in turn, is determined by the curvature of the smoothed latent log-likelihood. This is the subject of the following proposition.

Proposition D.3. *Suppose that, uniformly for $t \in [\tau, t_1]$, g is one-sided Lipschitz with constant ℓ_{τ,t_1} on the truncated region:*

$$\langle x - y, g(t,x) - g(t,y)\rangle \le \ell_{\tau,t_1}\|x-y\|^2 \tag{22}$$

Then Prop. D.1 applies with

$$\eta_\chi(s) = \beta(\kappa_s)\left(\lambda_\chi \ell_{\tau,t_1} - \rho_\chi\right). \tag{23}$$

Suppose, more strongly, that uniformly for $t \in [\tau, t_1]$, g is strongly dissipative with constant $\mu_{\tau,t_1} \ge 0$, on the truncated region:

$$\langle x - y, g(t,x) - g(t,y)\rangle \le -\mu_{\tau,t_1}\|x-y\|^2. \tag{24}$$

Then Prop. D.1 applies with

$$\eta_\chi(s) = -\left(\rho_\chi + \lambda_\chi \mu_{\tau,t_1}\right)\beta(\kappa_s). \tag{25}$$

Proof. Recalling the definition of the reverse-time drift F_χ from (17), it is straightforward to compute

$$\langle x - y, F_\chi(s,x) - F_\chi(s,y)\rangle = \lambda_\chi \beta(\kappa_s)\langle x - y, g(\kappa_s,x) - g(\kappa_s,y)\rangle - \rho_\chi\beta(\kappa_s)\|x-y\|^2.$$

The rates given in (23) and (25) now follow immediately after substituting (22) and (24), respectively. □

D.4 Guidance and discretisation error

The previous stability result controls the error between the exact guided dynamics and the exact approximately guided dynamics. It is also necessary to control the discretisation error introduced by a numerical solver.

Corollary D.4. *Assume that the hypotheses of Prop. D.1 hold for the exact and approximately guided ODE and SDE dynamics. Let $0 = s_0 < \cdots < s_N = R_{\tau,t_1}$ be a reverse-time grid with maximal step size h. Define*

$$A_{\eta_\chi} = \sup_{0\le s\le R_{\tau,t_1}} \Theta_{\eta_\chi}(s,0), \qquad H_{\eta_\chi,\delta_\chi} = \sup_{0\le s\le R_{\tau,t_1}} \int_0^s \Theta_{\eta_\chi}(s,r)\delta_\chi(r)\,\mathrm{d}r.$$

Suppose that the numerical approximations $\widehat{u}^{\leftarrow,h}$ and $\widehat{\xi}^{\leftarrow,h}$ to the approximately guided ODE and SDE satisfy

$$\max_{0\le n\le N}\|\widehat{u}^{\leftarrow}_{s_n} - \widehat{u}^{\leftarrow,h}_{s_n}\| \le C^{\mathrm{ode}}_{\mathrm{num},\tau,t_1}h^q, \qquad \Big\|\max_{0\le n\le N}\|\widehat{\xi}^{\leftarrow}_{s_n} - \widehat{\xi}^{\leftarrow,h}_{s_n}\|\Big\|_{L^p} \le C^{\mathrm{sde}}_{\mathrm{num},p}h^\varrho. \tag{26}$$

Then it holds that

$$\begin{aligned}\max_{0\le n\le N}\|u^{\leftarrow}_{s_n} - \widehat{u}^{\leftarrow,h}_{s_n}\| &\le A_{\eta_{\mathrm{ode}}}\|u^{\leftarrow}_0 - \widehat{u}^{\leftarrow}_0\| + H_{\eta_{\mathrm{ode}},\delta_{\mathrm{ode}}} + C^{\mathrm{ode}}_{\mathrm{num},\tau,t_1}h^q, \\ \Big\|\max_{0\le n\le N}\|\xi^{\leftarrow}_{s_n} - \widehat{\xi}^{\leftarrow,h}_{n}\|\Big\|_{L^p} &\le A_{\eta_{\mathrm{sde}}}\|\xi^{\leftarrow}_0 - \widehat{\xi}^{\leftarrow}_0\|_{L^p} + H_{\eta_{\mathrm{sde}},\delta_{\mathrm{sde}}} + C^{\mathrm{sde}}_{\mathrm{num},p}h^\varrho.\end{aligned} \tag{27}$$

In both displays, the initial-discrepancy term vanishes under common terminal coupling.

Proof. Apply Prop. D.1 at the grid points and take the maximum over n. For the ODE, add the deterministic numerical error by the triangle inequality. For the SDE, first use the pathwise stability bound, then take the L^p-norm and apply Minkowski's inequality together with (26). □

Corollary D.5. *Assume that the hypotheses of Corollary D.4 hold under a common terminal coupling, and consider a joint limit in which $S \to \infty$, $h \to 0$, and $a_S \to 0$. Suppose the realised guidance error satisfies $\varepsilon_{\tau,t_1} = O_{\mathbb{P}}(a_S)$ on $\mathcal{R}_{\tau,t_1}$. For $\chi \in \{\mathrm{ode}, \mathrm{sde}\}$, define*

$$B_{\chi,\tau,t_1} = \sup_{0\le s\le R_{\tau,t_1}} \int_0^s \Theta_{\eta_\chi}(s,r)\lambda_\chi \beta(\kappa_r)\,\mathrm{d}r.$$

Assume that the stability and numerical-error constants are uniformly controlled along this limit, in the sense that $B_{\chi,\tau,t_1} = O_{\mathbb{P}}(1)$, $C^{\mathrm{ode}}_{\mathrm{num},\tau,t_1} = O_{\mathbb{P}}(1)$, $C^{\mathrm{sde}}_{\mathrm{num},p} = O_{\mathbb{P}}(1)$. This is automatic when these quantities are deterministic and bounded independently of S and h. Then

$$\max_{0\le n\le N} \|u^{\leftarrow}_{s_n} - \widehat{u}^{\leftarrow,h}_{s_n}\| = O_{\mathbb{P}}(a_S + h^q), \qquad \max_{0\le n\le N} \|\xi^{\leftarrow}_{s_n} - \widehat{\xi}^{\leftarrow,h}_{n}\| = O_{\mathbb{P}}(a_S + h^\varrho).$$

In particular, if a uniform Monte Carlo guidance bound gives $a_S = S^{-1/2}$, then the corresponding rates are $O_{\mathbb{P}}(S^{-1/2} + h^q)$ for the ODE and $O_{\mathbb{P}}(S^{-1/2} + h^\varrho)$ for the SDE.

Proof. Under a common terminal coupling, the initial-discrepancy terms in Corollary D.4 vanish. Moreover,

$$H_{\eta_\chi,\delta_\chi} \le B_{\chi,\tau,t_1}\varepsilon_{\tau,t_1} = O_{\mathbb{P}}(a_S).$$

For the ODE, the numerical term satisfies $C^{\mathrm{ode}}_{\mathrm{num},\tau,t_1} h^q = O_{\mathbb{P}}(h^q)$, so the first conclusion follows from (27). For the SDE, the conditional L^p estimate in (26), together with $C^{\mathrm{sde}}_{\mathrm{num},p} = O_{\mathbb{P}}(1)$, implies that

$$\max_{0\le n\le N} \|\widehat{\xi}^{\leftarrow}_{s_n} - \widehat{\xi}^{\leftarrow,h}_{n}\| = O_{\mathbb{P}}(h^\varrho).$$

due to Markov's inequality. The second conclusion follows from the triangle inequality. □

D.5 Process-space propagation

The latent bounds pass directly to process space whenever the generator is Lipschitz on the region visited by the coupled latent trajectories.

Corollary D.6. *Assume the hypotheses of Corollary D.4. Suppose that E_m is equipped with a norm and that $T : \mathbb{R}^{m_\xi} \to E_m$ is L_T-Lipschitz on the latent region visited by the exact, approximately guided, and numerical trajectories. Then the ODE and SDE bounds in Corollary D.4 remain valid after applying T, with the right-hand sides multiplied by L_T. In particular,*

$$\max_{0\le n\le N} \|T(u^{\leftarrow}_{s_n}) - T(\widehat{u}^{\leftarrow,h}_{s_n})\| \le L_T\left(A_{\eta_{\mathrm{ode}}}\|u^{\leftarrow}_0 - \widehat{u}^{\leftarrow}_0\| + H_{\eta_{\mathrm{ode}},\delta_{\mathrm{ode}}} + C^{\mathrm{ode}}_{\mathrm{num},\tau,t_1}h^q\right),$$

$$\left\|\max_{0\le n\le N} \|T(\xi^{\leftarrow}_{s_n}) - T(\widehat{\xi}^{\leftarrow,h}_{n})\|\right\|_{L^p} \le L_T\left(A_{\eta_{\mathrm{sde}}}\|\xi^{\leftarrow}_0 - \widehat{\xi}^{\leftarrow}_0\|_{L^p} + H_{\eta_{\mathrm{sde}},\delta_{\mathrm{sde}}} + C^{\mathrm{sde}}_{\mathrm{num},p}h^\varrho\right).$$

Moreover, any Wasserstein bound above is inherited under pushforward, since $W_p(T_\#\mu, T_\#\nu) \le L_T W_p(\mu,\nu)$ whenever the relevant latent coupling is supported on the Lipschitz region.

Proof. The pathwise and strong L^p statements follow from $\|T(x) - T(y)\| \le L_T\|x - y\|$. The Wasserstein statement follows by pushing forward the same latent coupling through T. □

D.6 Additional path-law control for the reverse SDE

The preceding bounds control coupled paths. For the reverse-time SDE, one also obtains a path-law bound by Girsanov's theorem.

Proposition D.7. *Assume that the usual Girsanov conditions hold, and let P and $\widehat{\mathsf{P}}$ denote the path laws of the exact and approximately guided reverse SDEs on $C([0, R_{\tau,t_1}]; \mathbb{R}^{m_\xi})$. Assume also that $\mathcal{L}(\xi^{\leftarrow}_0) \ll \mathcal{L}(\widehat{\xi}^{\leftarrow}_0)$. Then*

$$\mathrm{KL}(\mathsf{P}\|\widehat{\mathsf{P}}) = \mathrm{KL}\Big(\mathcal{L}(\xi^{\leftarrow}_0)\,\Big\|\,\mathcal{L}(\widehat{\xi}^{\leftarrow}_0)\Big) + \frac{1}{2}\mathbb{E}_{\mathsf{P}}\int_0^{R_{\tau,t_1}} \beta(\kappa_s)\|g(\kappa_s,\xi^{\leftarrow}_s) - \widehat{g}(\kappa_s,\xi^{\leftarrow}_s)\|^2\,\mathrm{d}s. \quad (28)$$

In particular, if the two SDEs start from the same initial law, then the first term vanishes. If the realised guidance error is uniformly bounded by ε_{τ,t_1} along the exact paths, then, for $0 \le s \le R_{\tau,t_1}$,

$$\mathrm{KL}(\mathsf{P}_{[0,s]} \| \widehat{\mathsf{P}}_{[0,s]}) \le \mathrm{KL}\Big(\mathcal{L}(\xi_0^{\leftarrow}) \,\Big\|\, \mathcal{L}(\widehat{\xi}_0^{\leftarrow})\Big) + \frac{1}{2}\varepsilon_{\tau,t_1}^2 \int_0^s \beta(\kappa_r)\,\mathrm{d}r.$$

Consequently,

$$\Big\|\mathcal{L}(\xi_s^{\leftarrow}) - \mathcal{L}(\widehat{\xi}_s^{\leftarrow})\Big\|_{\mathrm{TV}}^2 \le \frac{1}{2}\mathrm{KL}\Big(\mathcal{L}(\xi_0^{\leftarrow}) \,\Big\|\, \mathcal{L}(\widehat{\xi}_0^{\leftarrow})\Big) + \frac{1}{4}\varepsilon_{\tau,t_1}^2 \int_0^s \beta(\kappa_r)\,\mathrm{d}r.$$

Proof. The two SDEs have a common diffusion coefficient. Moreover, we have from the definitions that

$$\|\Sigma_{\mathrm{sde}}(s)^{-1}(b(s,u) - \widehat{b}(s,u))\|^2 = \beta(\kappa_s)\|g(\kappa_s,u) - \widehat{g}(\kappa_s,u)\|^2.$$

The identity in (28) follows immediately from Girsanov's theorem. The KL bound follows from the uniform guidance-error assumption. The total-variation bound follows by data processing and Pinsker's inequality. □

D.7 Derivative-free guidance

When $\nabla_\xi \log \Lambda(\xi)$ is unavailable, one can estimate the guidance field using a form of Fisher's identity. Under the tilted bridge conditional density, we have

$$p(\xi_0 \mid \xi_t = u, \mathcal{C}) \propto \Lambda(\xi_0)\mathcal{N}\big(\xi_0; \alpha(t)u, \sigma(t)^2 I_{m_\xi}\big)\,.$$

Using Fisher's identity, it follows that

$$\nabla_u \log \rho_t(u) = -\frac{1}{\sigma(t)^2}\left(u - \alpha(t)\mathbb{E}[\xi_0 \mid \xi_t = u, \mathcal{C}]\right).$$

Meanwhile, since $\rho_t(u) \propto \psi(t,u)r(u)$, we have $\nabla_u \log \rho_t(u) = g(t,u) - u$. Combining this with the previous expression, we arrive at

$$g(t,u) = \frac{\alpha(t)}{\sigma(t)^2}\left(\mathbb{E}[\xi_0 \mid \xi_t = u, \mathcal{C}] - \alpha(t)u\right).$$

This immediately suggests a gradient-free estimator. For a given pair (t,u), draw bridge samples

$$\xi^{(s)} = \alpha(t)u + \sigma(t)\epsilon^{(s)}, \qquad \epsilon^{(s)} \sim \mathcal{N}(0, I_{m_\xi}), \qquad s = 1, \dots, S.$$

We can then define the self-normalised approximation

$$\widehat{g}_S^{\mathrm{df}}(t,u) = \frac{\alpha(t)}{\sigma(t)^2}\left(\sum_{s=1}^S \bar{w}_s \xi^{(s)} - \alpha(t)u\right), \qquad \bar{w}_s = \frac{\Lambda(\xi^{(s)})}{\sum_{\ell=1}^S \Lambda(\xi^{(\ell)})}.$$

This estimator only requires likelihood evaluations, but can be less sample-efficient than the gradient-based estimator because it estimates a small conditional-mean difference. Moreover, the prefactor $\alpha(t)/\sigma(t)^2$ can amplify Monte Carlo noise at small noising times.

E Fully Bayesian inference for model parameters

In this appendix we provide a measure-theoretic statement of the augmented construction in Sec. 3, mirroring the results in App. B, followed by additional details regarding the guidance.

E.1 Proof of the Latent Conditioning Principle in the Augmented Latent Space

Let $(E_m, \mathscr{E}_m)$ be the discretised process space and $(\Theta, \mathcal{T})$ a measurable parameter space. Let Q be a prior probability measure on Θ, and let P_ϑ denote the prior law of f_0 conditional

on ϑ. We assume that $\vartheta \mapsto P_\vartheta$ is a probability kernel from $(\Theta, \mathcal{T})$ to $(E_m, \mathscr{E}_m)$. The joint prior law of (f_0, ϑ) is then

$$P(\mathrm{d}f_0, \mathrm{d}\vartheta) = P_\vartheta(\mathrm{d}f_0)Q(\mathrm{d}\vartheta).$$

For a measurable likelihood $L_{\mathcal{C}} : E_m \times \Theta \to [0, \infty)$, define

$$\Pi(\mathrm{d}f_0, \mathrm{d}\vartheta \mid \mathcal{C}) = \frac{L_{\mathcal{C}}(f_0, \vartheta)P_\vartheta(\mathrm{d}f_0)Q(\mathrm{d}\vartheta)}{\widetilde{Z}}, \qquad \widetilde{Z} = \int_\Theta \int_{E_m} L_{\mathcal{C}}(f_0, \vartheta)P_\vartheta(\mathrm{d}f_0)Q(\mathrm{d}\vartheta), \tag{29}$$

whenever $0 < \widetilde{Z} < \infty$. This includes the case in which the likelihood depends on ϑ only through the generated process value f_0.

Let $R_\xi = \mathcal{N}(0, I_{m_\xi})$ and $R_\zeta = \mathcal{N}(0, I_{m_\vartheta})$. Suppose there is a measurable prior generator $H : \mathbb{R}^{m_\vartheta} \to \Theta$ satisfying $H_\# R_\zeta = Q$, and that, for each $\vartheta \in \Theta$, the process generator $T_\vartheta : \mathbb{R}^{m_\xi} \to E_m$ satisfies $(T_\vartheta)_\# R_\xi = P_\vartheta$. Assume also that $(\xi, \vartheta) \mapsto T_\vartheta(\xi)$ is jointly measurable. Define

$$\widetilde{R} = R_\xi \otimes R_\zeta, \qquad \widetilde{T}(\xi, \zeta) = \big(T_{H(\zeta)}(\xi), H(\zeta)\big),$$

and the augmented latent target

$$\widetilde{\rho}(\mathrm{d}\xi, \mathrm{d}\zeta \mid \mathcal{C}) = \frac{L_{\mathcal{C}}\big(T_{H(\zeta)}(\xi), H(\zeta)\big)R_\xi(\mathrm{d}\xi)R_\zeta(\mathrm{d}\zeta)}{\widetilde{Z}}. \tag{30}$$

Proposition E.1. *Under the assumptions above,*

$$\widetilde{T}_\# \widetilde{R} = P, \qquad \widetilde{T}_\# \widetilde{\rho}(\cdot \mid \mathcal{C}) = \Pi(\cdot \mid \mathcal{C}).$$

In particular, if $(\xi_0, \zeta_0) \sim \widetilde{\rho}(\cdot \mid \mathcal{C})$, then $\vartheta = H(\zeta_0)$ has the Bayesian parameter posterior and $f_0 = T_{H(\zeta_0)}(\xi_0)$ is a draw from the corresponding joint conditional process law.

Proof. Let $A \in \mathscr{E}_m \otimes \mathcal{T}$ be measurable. Using first the definition of pushforward and then $(T_\vartheta)_\# R_\xi = P_\vartheta$ and $H_\# R_\zeta = Q$, we obtain

$$\begin{aligned}
\widetilde{T}_\# \widetilde{R}(A) &= \int \mathbb{1}_A\big(T_{H(\zeta)}(\xi), H(\zeta)\big)R_\xi(\mathrm{d}\xi)R_\zeta(\mathrm{d}\zeta) \\
&= \int_\Theta \int_{E_m} \mathbb{1}_A(f_0, \vartheta)P_\vartheta(\mathrm{d}f_0)Q(\mathrm{d}\vartheta) \\
&= P(A).
\end{aligned}$$

The same calculation after likelihood tilting gives

$$\begin{aligned}
\widetilde{T}_\# \widetilde{\rho}(A \mid \mathcal{C}) &= \frac{1}{\widetilde{Z}} \int \mathbb{1}_A\big(T_{H(\zeta)}(\xi), H(\zeta)\big)L_{\mathcal{C}}\big(T_{H(\zeta)}(\xi), H(\zeta)\big)R_\xi(\mathrm{d}\xi)R_\zeta(\mathrm{d}\zeta) \\
&= \frac{1}{\widetilde{Z}} \int_\Theta \int_{E_m} \mathbb{1}_A(f_0, \vartheta)L_{\mathcal{C}}(f_0, \vartheta)P_\vartheta(\mathrm{d}f_0)Q(\mathrm{d}\vartheta) \\
&= \Pi(A \mid \mathcal{C}).
\end{aligned}$$

This proves both claims. □

Corollary E.2. *Suppose that the regularity assumptions of Theorem C.2 hold for the augmented latent target in (30). Then the exact reverse-time LatentFlow SDE, applied in $\mathbb{R}^{m_\xi + m_\vartheta}$, produces a terminal draw $(\xi_0, \zeta_0) \sim \widetilde{\rho}(\cdot \mid \mathcal{C})$, and its image under $\widetilde{T}$ has joint posterior law $\Pi(\cdot \mid \mathcal{C})$. The analogous statement holds for the probability-flow ODE under the assumptions of Theorem C.3.*

Proof. Apply Theorems C.2 and C.3 to the single augmented generator $\widetilde{T}$ and then use Prop. E.1. □

Marginalising (29) over f_0 gives

$$\Pi(\mathrm{d}\vartheta \mid \mathcal{C}) = \frac{Z_\vartheta Q(\mathrm{d}\vartheta)}{\widetilde{Z}}, \qquad Z_\vartheta = \int_{E_m} L_{\mathcal{C}}(f_0, \vartheta)P_\vartheta(\mathrm{d}f_0).$$

Thus the fully Bayesian construction averages over parameter values according to their posterior probability. Importantly, LatentFlow does not need to evaluate Z_ϑ: it samples the joint target using only pointwise evaluations of the pulled-back likelihood.

E.2 Guidance in the augmented latent space

We now return to the differentiable density setting of the main text. Suppose we now write

$$\ell_{\mathcal{C}}(f_0, \vartheta) = \log L_{\mathcal{C}}(f_0, \vartheta), \qquad \vartheta = H(\zeta), \qquad f_0 = T_\vartheta(\xi).$$

When H, T_ϑ, and $L_{\mathcal{C}}$ are differentiable and $L_{\mathcal{C}} > 0$, the score of the augmented latent likelihood is given by

$$\nabla_\xi \log \widetilde{\Lambda}_{\mathcal{C}}(\xi, \zeta) = [D_\xi T_\vartheta(\xi)]^\top \nabla_{f_0} \ell_{\mathcal{C}}(f_0, \vartheta), \tag{31}$$

$$\nabla_\zeta \log \widetilde{\Lambda}_{\mathcal{C}}(\xi, \zeta) = [DH(\zeta)]^\top \left\{ [D_\vartheta T_\vartheta(\xi)]^\top \nabla_{f_0} \ell_{\mathcal{C}}(f_0, \vartheta) + \nabla_\vartheta \ell_{\mathcal{C}}(f_0, \vartheta) \right\}, \tag{32}$$

where the right-hand sides are evaluated at $\vartheta = H(\zeta)$ and $f_0 = T_\vartheta(\xi)$. The first line guides the process innovation, while the second guides the parameter innovation through both the parameter dependence of the process generator and any explicit parameter dependence of the likelihood. Naturally, if the likelihood $L_{\mathcal{C}}$ depends on ϑ only through f_0, the final term in (32) vanishes.

In practice, guidance terms in (31) and (32) are of course estimated using Monte Carlo samples. For a noised augmented state $\widetilde{u}$, draw

$$\widetilde{\varepsilon}^{(s)} \sim \mathcal{N}(0, I_{m_\xi + m_\vartheta}), \qquad \widetilde{\xi}_{0|t}^{(s)} = \alpha(t)\widetilde{u} + \sigma(t)\widetilde{\varepsilon}^{(s)}, \qquad s = 1, \dots, S.$$

The Monte Carlo guidance estimator is then precisely the analogue of (4). In particular, we now set

$$\widehat{\widetilde{g}}_S(t, \widetilde{u}) = \alpha(t) \sum_{s=1}^{S} \widetilde{w}_s \nabla_{\widetilde{\xi}} \log \widetilde{\Lambda}_{\mathcal{C}}\big(\widetilde{\xi}_{0|t}^{(s)}\big), \qquad \widetilde{w}_s = \frac{\widetilde{\Lambda}_{\mathcal{C}}\big(\widetilde{\xi}_{0|t}^{(s)}\big)}{\sum_{\ell=1}^{S} \widetilde{\Lambda}_{\mathcal{C}}\big(\widetilde{\xi}_{0|t}^{(\ell)}\big)}.$$

Consequently, the same bridge candidates and likelihood evaluations jointly guide the process and its parameters.

For common continuous priors, the existence of the prior generator H is immediate. For example, an affine map gives a Gaussian prior; exponentiation or a distributional quantile map gives a positive prior; a quantile map gives a bounded prior; and a logistic map gives a simplex-valued prior. Meanwhile, hierarchical priors can be handled by concatenating the underlying independent innovations and composing their conditional generators. Discrete parameters also admit measurable prior generators, so Prop. E.1 remains valid, but the resulting pulled-back likelihood is generally non-differentiable; in that case one may use a relaxation, a surrogate, or the derivative-free guidance methods in Sec. D.7.

F Additional Related Work

Conditioning stochastic-process priors. Perhaps the most well-studied case of stochastic-process conditioning is inference for Gaussian processes (GPs). Under finitely many linear-Gaussian observations, the posterior over any finite collection of function values remains Gaussian and can be computed analytically (Rasmussen & Williams, 2006). This conjugacy is generally lost under non-Gaussian likelihoods, motivating a large literature on approximate inference (Kuss & Rasmussen, 2005; Nickisch & Rasmussen, 2008; Hensman et al., 2015; Nickisch et al., 2018). Related work encodes known linear operator or differential-equation constraints directly in the GP prior or regression construction (Jidling et al., 2017; Lange-Hegermann, 2018; Gulian et al., 2022). By contrast, shape information such as monotonicity, boundedness, positivity, and convexity is commonly incorporated through virtual inequality observations, finite-dimensional coefficient inequalities, truncation, or posterior projection, generally resulting in non-Gaussian or projected posterior laws (Riihimäki & Vehtari, 2010; Da Veiga & Marrel, 2012; Lin & Dunson, 2014; Maatouk & Bay, 2017; López-Lopera et al., 2018; Agrell, 2019; Swiler et al., 2020; Maatouk et al., 2025; Astfalck et al., 2018). An alternative generalised Bayes linear approach enforces domain restrictions through the choice of inferential solution space (Astfalck et al., 2024). The closest GP-specific antecedent to the current work is FlowGP (Moss et al., 2026), which conditions a finite-dimensional whitened GP representation on nonlinear or non-Gaussian information by solving a guided

probability-flow ODE. Our framework contains FlowGP as a special case but applies to a substantially broader class of stochastic-process priors; see App. A.

For SDE priors, LatentFlow is closely connected to diffusion-bridge simulation and conditioned diffusion processes. Classical bridge methods condition a diffusion to hit specified endpoints or endpoint observations, often by adding an approximate guiding term to its drift. Delyon & Hu (2006) construct readily simulated endpoint-guided diffusions and derive an explicit likelihood-ratio correction. Schauer et al. (2017) derive guiding terms from auxiliary tractable diffusions with known transition densities, while Bierkens et al. (2020) extend this guided-proposal framework to partial endpoint observations and elliptic or hypoelliptic settings. Related bridge samplers use coupling or other proposal mechanisms (Bladt & Sørensen, 2014; Bladt et al., 2016). More recent methods learn time reversals using score matching (Heng et al., 2025), estimate bridge scores without first learning the time reversal (Baker et al., 2025), develop infinite-dimensional conditioning constructions (Baker et al., 2024), introduce neural guided bridges (Yang et al., 2025b), or exploit Malliavin-calculus identities (Pidstrigach et al., 2025). Rare-event simulation and transition-path sampling address closely related path ensembles. Transition-path sampling uses path-space Monte Carlo to study rare reactive trajectories (Dellago et al., 1998; 2002), with recent diffusion-based methods also targeting transition-path ensembles (Seong et al., 2025), whereas genealogical particle methods and adaptive multilevel splitting estimate rare-event probabilities by repeatedly selecting and resampling trajectories according to their progress toward a rare set (Del Moral & Garnier, 2005; Cérou & Guyader, 2007; Cérou et al., 2012). LatentFlow targets likelihood-tilted path laws by tilting the law of the underlying innovations and transporting it toward regions of high condition likelihood. It therefore provides a likelihood-guided complement to output-space bridge proposals and splitting methods when suitable pulled-back gradients or guidance estimators are available.

Latent and reference-space formulations. Many stochastic-process laws admit representations as pushforwards of a tractable reference distribution rather than through an explicit density on the resulting process space. This perspective is implicit in simulator and filtering representations of state-space models (Doucet et al., 2001; Särkkä, 2013); in GMRF and SPDE constructions of spatial fields (Rue & Held, 2005; Lindgren et al., 2011); and in process priors and finite-dimensional representations built by composing or transforming latent random variables or functions, including random-feature kernel representations, latent-force models, GP state-space models, and deep GPs (Rahimi & Recht, 2007; Álvarez et al., 2009; 2013; Frigola et al., 2013; Damianou & Lawrence, 2013; Frigola et al., 2014). Pathwise GP constructions based on Matheron's rule similarly represent posterior sample paths as transformations of prior samples and provide scalable samples for downstream Monte Carlo computation (Wilson et al., 2020; 2021). Innovation and non-centred parameterisations make this explicit for diffusion processes by parameterising latent paths through transformed path variables or driving-noise variables together with model parameters (Roberts & Stramer, 2001; Golightly & Wilkinson, 2008; Papaspiliopoulos et al., 2013; Whitaker et al., 2017; Graham et al., 2022). Neural-process and implicit-process families likewise define learned distributions over functions through neural maps (Garnelo et al., 2018; Kim et al., 2019; Ma et al., 2019; Gordon et al., 2020). Similarly, normalising flows push forward simple latent laws, although they usually learn invertible transformations so that output densities remain available by change of variables (Rezende & Mohamed, 2015; Papamakarios et al., 2021).

The same reference-law viewpoint is central to Bayesian inverse problems. A posterior on a possibly infinite-dimensional parameter or function space is commonly defined by tilting a tractable reference measure by a likelihood (Stuart, 2010; Dashti & Stuart, 2017). Function-space MCMC methods, including pCN and Hilbert-space HMC, are designed directly for such targets and can remain well behaved under mesh refinement (Beskos et al., 2011; Cotter et al., 2013; Hairer et al., 2014). SMC, annealed importance sampling, and flow-augmented transport samplers construct sequences of intermediate distributions between reference and posterior laws (Neal, 2001; Del Moral et al., 2006; Arbel et al., 2021), while transport-map methods seek explicit reference-to-posterior maps, including in function-space inverse problems (El Moselhy & Marzouk, 2012; Hosseini et al., 2025). Recent score-based methods have likewise emphasised defining and approximating posterior sampling dynamics directly on function spaces (Baldassari et al., 2023; 2024; 2025).

Bayesian inference in generator noise space. A related literature performs posterior inference directly in the input-noise coordinates of a fixed generative model. For example, Holden et al. (2022) use parallel-tempered pCN in the latent space of a VAE, while Patel et al. (2022) use HMC together with a GAN. Another early contribution is Böhm et al. (2019). Normalising-flow priors have also been combined with variationally trained conditional flows for amortised inverse-problem inference (Whang et al., 2021). Related early work performs constrained-HMC inference in the input-noise space of a differentiable generator (Graham & Storkey, 2017), while distributional-control methods learn a transformation of the generator's reference law toward specified output-space constraints (Wu et al., 2022).

There is also a substantial literature on combining transport maps with Monte Carlo methods. Transport-map methods construct invertible transformations between tractable reference laws and target laws, either as approximate posterior representations or as preconditioners and proposals for Monte Carlo sampling (El Moselhy & Marzouk, 2012; Parno & Marzouk, 2018; Hoffman et al., 2019; Cabezas & Nemeth, 2023; Cabezas et al., 2024). When an approximate map is embedded within a valid MCMC kernel with the appropriate correction, map approximation error affects sampling efficiency rather than the invariant target (Parno & Marzouk, 2018; Hoffman et al., 2019). NeuTra HMC uses a variationally trained normalising flow followed by HMC in the transported coordinates (Hoffman et al., 2019), whereas transport elliptical slice sampling uses a related nonlinear preconditioning strategy for elliptical slice sampling (Cabezas & Nemeth, 2023). A closely related latent construction is the neural-transport MCMC method of Nijkamp et al. (2022), in which an energy correction exponentially tilts the output law of a learned flow and sampling is performed in its latent coordinates. That work is directed toward learning and sampling energy-based generative models, rather than conditioning a stochastic-process simulator.

More recent methods apply the same principle to diffusion and flow-based generators. Outsourced diffusion sampling considers a deterministic representation $x = f_\vartheta(z)$, with z drawn from a Gaussian reference, and trains an amortised diffusion sampler for the latent posterior induced by an output-space likelihood or reward (Venkatraman et al., 2025). Noise-space HMC instead applies HMC directly to the initial noise of a deterministic reverse-diffusion map (Xia et al., 2026). Similar constructions have appeared for flow-matching models: Source-Guided Flow Matching modifies the source distribution while retaining the learned vector field, whereas D-Flow SGLD samples the source posterior induced by a new measurement operator (Wang et al., 2026; Parikh et al., 2026). Related inverse-problem methods perform SMC in the latent coordinates of diffusions, guide sampling trajectories under pre-trained latent flow priors, or combine latent diffusion posterior sampling with surrogate-likelihood guidance (Achituve et al., 2025; Askari et al., 2025; Wang & Tartakovsky, 2026).

These approaches share the central principle of conditioning a generator by changing the distribution of its input noise. LatentFlow differs primarily in the scope of the generator and in how the latent posterior is sampled. In our case, T_ϑ may describe an explicit stochastic-process prior, a numerical simulator, or a pre-trained neural generator, and need not be invertible or density-evaluable. Moreover, LatentFlow does not train an auxiliary conditional generator or amortised latent sampler. Instead, its time-dependent guidance is determined directly by the pulled-back likelihood and a known Gaussian noising bridge.

Diffusion guidance and finite-time controlled transport. The sampling construction underlying LatentFlow is closely related to diffusion and flow-based generative modelling. Diffusion probabilistic models define a forward noising process from a data distribution to a simple reference law together with a learned reverse process (Sohl-Dickstein et al., 2015; Ho et al., 2020); continuous-time score-based models generate samples by reversing a noising SDE or solving the associated probability-flow ODE (Song et al., 2021). Flow matching, rectified flows, and stochastic interpolants provide related continuous-time transport formulations (Lipman et al., 2023; Liu et al., 2023b; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025). Recent guided flow-matching methods extend these constructions to conditional generation and energy- or reward-guided sampling (Feng et al., 2025; Mark et al., 2025).

A large literature conditions pre-trained diffusion models on labels, measurements, inverse problems, or reward functions by modifying their reverse dynamics through classifier, classifier-

free, measurement, reward, or off-the-shelf guidance terms (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Chung et al., 2023; Bansal et al., 2024). To control bias introduced by approximate guidance, other conditional diffusion samplers use SMC, particle MCMC, Feynman–Kac representations, twisting, or related correction mechanisms (Wu et al., 2023; Cardoso et al., 2024; Corenflos et al., 2025; Singhal et al., 2025; Skreta et al., 2025; Wu et al., 2025). Under their stated assumptions, some of these methods are asymptotically exact as the particle number increases, while others preserve a model-defined conditional target through a valid Monte Carlo correction; see Zhao et al. (2025) for a recent review. Diffusion and flow models have also been developed directly for function-valued data and function-space posterior sampling (Dutordoir et al., 2023; Franzese et al., 2023; Kerrigan et al., 2024; Lim et al., 2025; Yao et al., 2025).

These constructions admit a broader interpretation through finite-time stochastic control. Doob h-transforms provide a classical mechanism for conditioning Markov processes; for diffusions, the transformed dynamics acquire a drift correction involving the log-gradient of an appropriate space–time h-function (Doob, 1957; Didi et al., 2023). Related entropy-minimisation and Schrödinger-bridge formulations characterise controlled path measures as minimum-relative-entropy perturbations of a reference process subject to marginal or endpoint constraints (Föllmer, 1985; Léonard, 2014; Chen et al., 2016). Modern Schrödinger-bridge, Schrödinger–Föllmer, and controlled-diffusion samplers exploit this finite-time transport perspective for sampling unnormalised targets (Bernton et al., 2019; Huang et al., 2021; Zhang & Chen, 2022; Vargas et al., 2023b;a; 2024; Tamogashev & Malkin, 2026), while diffusion Schrödinger-bridge methods construct transports between prescribed endpoint laws (De Bortoli et al., 2021; Peluchetti, 2023; Liu et al., 2023a; 2024). Several recent conditional-diffusion methods make the connection to Doob transforms explicit (Didi et al., 2023; Denker et al., 2024; Guo et al., 2026). In infinite dimensions, Park et al. (2024) develop a function-space stochastic-control formulation of diffusion bridges; Yang et al. (2025a) learn discretisation-equivariant bridge scores using operator learning; and Pieper-Sethmacher et al. (2025) construct guided bridge measures for semilinear SPDEs conditioned on linear terminal observations. Baker et al. (2026) develop a complementary construction for conditioning trained function-space diffusion priors by supervised guidance.

LatentFlow instantiates the finite-time guidance principle in latent innovation space. Rather than learning a noising or transport process from samples, it uses a known reference-space bridge and the stochastic-process law supplied by T_ϑ. Its guiding field is the gradient of $\log h_t$, where h_t is the conditional expectation of the pulled-back likelihood under the latent noising bridge, equivalently the log-gradient of the corresponding smoothed pulled-back likelihood. Thus, LatentFlow combines the pullback formulation used in generator-noise inference with a diffusion-style finite-time transport, yielding a reference-space sampler for a broad class of stochastic-process priors.

G Additional Experimental Details and Numerical Results

In this appendix we provide full experimental details for the results in Sec. 5, as well as some additional numerical results.

G.1 Sampling Benchmarks

We first evaluate LatentFlow on three canonical benchmark distributions that are challenging for standard inference methods. Each target takes the form $p^*(x) \propto p_{\text{prior}}(x) \cdot p(y \mid x)$, with prior $p_{\text{prior}}(x) = \mathcal{N}(x; 0, I)$ and likelihoods $p(y \mid x)$ described below.

For LatentFlow, we use an Euler–Maruyama discretisation with 100 time steps, and 10 Monte Carlo samples for the guidance approximation. We benchmark against the No-U-Turn Sampler (NUTS; Hoffman & Gelman, 2014), mean-field VI (Blei et al., 2017), and a Laplace approximation. Sample quality is measured using the Maximum Mean Discrepancy (MMD^2) with a multi-scale RBF kernel, with uncertainty reported as standard errors over 5 independent runs, alongside the effective sample size per second (ESS/s). We generate

$N = 1{,}000$ samples using each method. NUTS thus requires $1,000$ steps to produce its $1,000$ samples, as well as an additional 500 warmup steps to ensure the chains are well mixed.

Across all benchmarks, LatentFlow achieves MMD^2 scores competitive with or better than NUTS, while producing *independent* samples. LatentFlow also succeeds on problems where standard MCMC collapses to a single mode.

Banana Distribution. The banana (twisted Gaussian) distribution has a curved, parabolic valley that resists random-walk exploration:

$$x_1 \sim \mathcal{N}(0, \sigma_1^2), \qquad x_2 \mid x_1 \sim \mathcal{N}\big(b(x_1^2 - \sigma_1^2), \sigma_2^2\big),$$

with $b = 0.7$, $\sigma_1 = 1.0$, and $\sigma_2 = 0.4$. Fig. 7 shows samples from each method overlaid on the true density. LatentFlow closely follows the curved valley, while VI and Laplace fail to capture the non-linear geometry. NUTS tracks the true distribution but produces far fewer independent samples, reflected in the low ESS of 67; see Tab. 2. LatentFlow achieves the best MMD^2, and is substantially faster than NUTS, achieving over 100 times greater ESS/s.

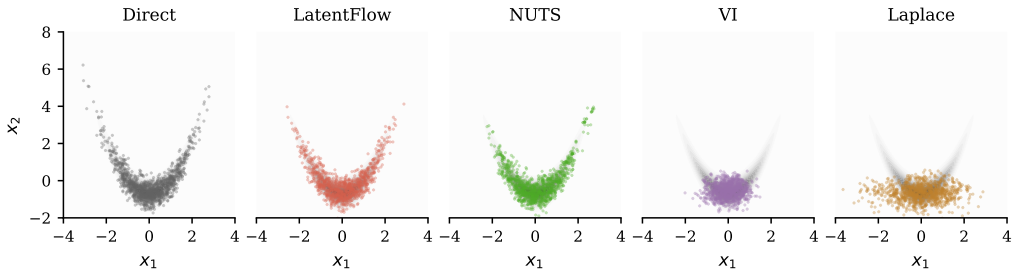


Figure 7: Banana distribution. Background: true density. Scatter: 1,000 samples from each method. Direct samples are exact draws. LatentFlow matches the true density well. VI and Laplace fail to capture the curved geometry.

Table 2: Banana distribution. $\text{MMD}^2 \pm$ stderr (lower is better), wall-clock time, and ESS/s, all over 5 independent runs (3 d.p.).

Method	**MMD^2**	**Time (s)**	**ESS**	**ESS/s**
LatentFlow	-0.001 ± 0.000	0.060 ± 0.000	1000	18010.0
NUTS	0.004 ± 0.003	0.470 ± 0.160	67	141.7
VI	0.059	0.310	1000	3270.7
Laplace	0.044	—	—	—

Ring Distribution. The ring (shell) distribution is concentrated on a thin annulus of radius R, with density given by

$$p^*(x) \;\propto\; \exp\left(-\frac{(\|x\|^2 - R^2)^2}{2\sigma^2}\right),$$

with $R = 2.0$ and $\sigma = 0.3$. Since gradients point predominantly *radially*, MCMC chains circulate slowly around the ring and mix poorly. As shown in Fig. 8, LatentFlow traces the full annulus accurately and achieves over $20\times$ the ESS/s of NUTS.

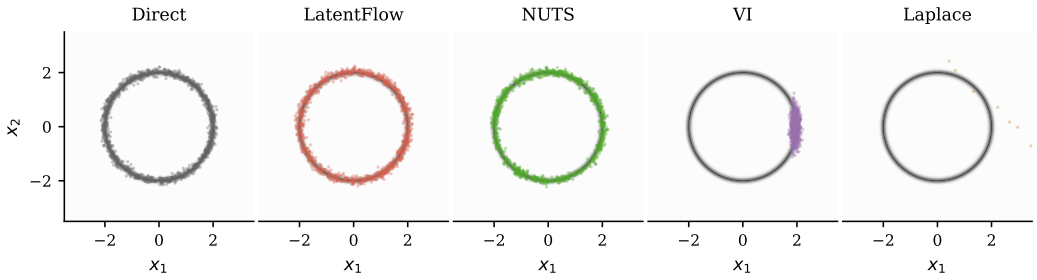


Figure 8: Ring distribution. Background: true density. Scatter: 1,000 samples from each method. LatentFlow traces the annulus accurately. NUTS also covers the ring but requires more warm-up steps. VI collapses to a small part of the ring. Laplace diverges catastrophically.

Table 3: Ring distribution. $\text{MMD}^2 \pm$ stderr (lower is better), wall-clock time, ESS/s, and radial statistics, all over 5 independent runs (3 d.p.). Ground truth: $\mathbb{E}[r] = 2$, $\text{Std}[r] \approx 0.076$.

Method	**MMD^2**	**Time (s)**	**ESS/s**	$\mathbb{E}[r]$	Std$[r]$
LatentFlow	0.000 ± 0.000	0.050 ± 0.000	21263.4	2.004	0.086
NUTS	0.001 ± 0.001	0.340 ± 0.060	897.1	1.999	0.075
VI	0.453	0.290	3412.0	1.960	0.087
Laplace	0.458	—	—	274.105	209.312

Mixture of Gaussians. We test multi-modal coverage using a mixture of $K = 4$ equal-weight Gaussians centred at $(\pm 2, \pm 2)$ with $\sigma = 0.5$. LatentFlow places samples in all four quadrants, while NUTS concentrates in the upper-right quadrant; see Fig. 9. While NUTS achieves a raw ESS of 943 and an ESS/s of 2,191.4 (Tab. 4), its samples originate from a single mode. Its apparent efficiency is thus misleading: it fails to cover multiple modes, reflected in its MMD^2 of 0.401 ± 0.009. LatentFlow achieves the best sample quality with $\text{MMD}^2 = 0.00079 \pm 0.00027$, recovering the correct global mean of approximately $(0, 0)$ and standard deviation near 2.04 in both dimensions, compared to NUTS's collapsed mean of (2.016, 1.984) with standard deviation ≈ 0.490.

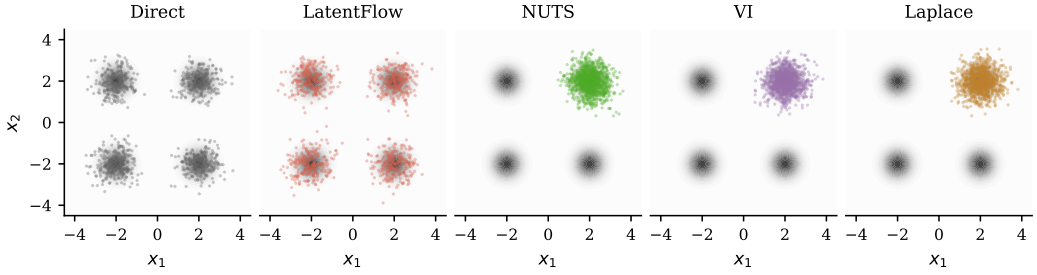


Figure 9: Mixture of Gaussians. Background: true density (four modes). Scatter: 1,000 samples from each method. LatentFlow covers all four modes. NUTS collapses entirely to quadrant 4 (100% of the chain). VI collapses to a single Gaussian centred near $(2, 2)$.

Table 4: Mixture of Gaussians. $\text{MMD}^2 \pm$ stderr (lower is better), wall-clock time, and ESS/s, all over 5 independent runs (3 d.p.). Despite its high ESS/s, NUTS fails to achieve multi-modal coverage (all samples are from one mode).

Method	**MMD^2**	**Time (s)**	**ESS**	**ESS/s**
LatentFlow	0.001 ± 0.000	0.050 ± 0.000	1000	19865.9
NUTS	0.401 ± 0.009	0.430 ± 0.130	943	2191.4
VI	0.413	0.140	1000	7119.4

G.2 Spatial Processes

We next provide full experimental details and additional results for the spatial processes described in Sec. 5. We consider three priors of increasing departure from Gaussianity: a *Gaussian process*, a *Student-t process*, and a *Cauchy convolution* field. The conditioning statements correspond to spares point observations, regional summaries (areal max and areal mean), and censored observations. Posterior samples are produced by integrating the LatentFlow SDE, using a single Monte Carlo reconstruction per step ($S = 1$) and 500 integration steps.

G.2.1 Priors

All priors are realised as deterministic, differentiable push-forward maps of a standard-Gaussian latent vector, $x = g(z)$ with $z \sim \mathcal{N}(0, I)$, on the 64×64 lattice.

Gaussian process. A smooth Gaussian field obtained from a Matérn-3/2 covariance with lengthscale 0.15.

Student-t process. A heavy-tailed, spatially coupled field with $\nu_t = 4$ degrees of freedom and a broader Matérn-3/2 lengthscale of 0.5. The shared scale variable couples the tails across the whole domain, so large excursions tend to occur field-wide rather than at isolated pixels.

Cauchy convolution. A very heavy-tailed field constructed by convolving a Cauchy noise source with a short-lengthscale ($\ell = 0.05$) kernel. The infinite-variance source produces sharp, highly localised, high-contrast features superimposed on a smoother background.

G.2.2 Conditions

We condition each prior on four different conditions, specified by the following four observation operators. Three of these are equality constraints, enforced through a Gaussian-likelihood relaxation $\log p(y \mid x) = -\|r(x)\|^2/(2\sigma^2)$ with a small tolerance $\sigma = 10^{-5}$ on the residual $r(x)$. The final one (censoring) is a one-sided inequality.

Sparse observations. Exact values are observed at $N = 30$ uniformly sampled pixels. The residual is the difference between the sampled field and the observed values at those sites, $r(x) = x_{\mathcal{S}} - y_{\mathcal{S}}$.

Areal maximum. The domain is partitioned into 8 Voronoi regions. For each region R we condition on the true maximum of a reference draw over that region, $m_R = \max_{i \in R} x_i^{\text{ref}}$. The residual has two parts: a hinge penalty $\sum_{i \in R} \max(x_i - m_R, 0)$ forbidding any pixel from exceeding the regional target, and a soft maximum term that drives the attained regional maximum up to m_R. This enforces the condition that the largest value in region R equals m_R, without requiring which pixel attains it.

Areal mean. The domain is partitioned into a finer tiling of 20 Voronoi regions. We condition on the true area average of the reference draw in each region, $\bar{x}_R = |R|^{-1} \sum_{i \in R} x_i^{\text{ref}}$. The residual is the vector of per-region mean discrepancies.

Censored observations. We observe whether the field lies above or below a site-specific detection limit (left- or right-censoring determined by the reference field relative to its median) at $N = 45$ sites. Each site contributes a one-sided hinge residual that is zero whenever the sample lies on the correct side of the limit (with a small margin of 0.2) and grows linearly with the violation.

G.2.3 Results

Below we provide the results for each of these spatial process priors and conditions. To be specific, each figure shows six samples from the prior and the posteriors corresponding to each of the conditions.

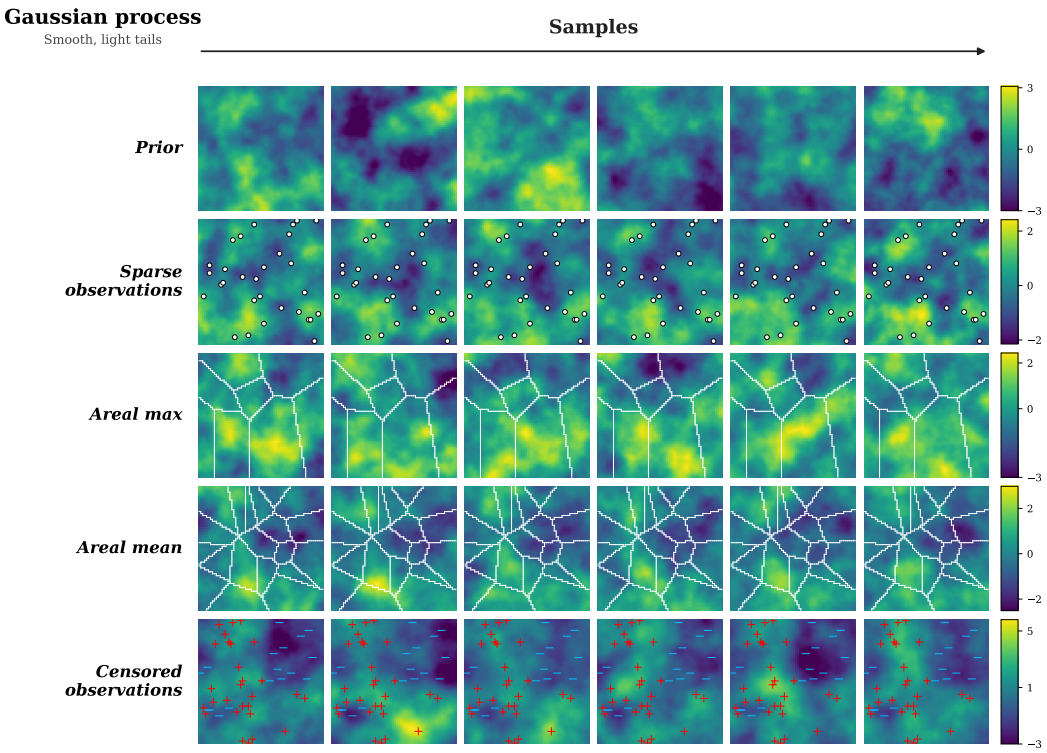


Figure 10: **Conditional sampling from a Gaussian process prior.** The base process is a smooth, light-tailed Gaussian field (Matérn-3/2 kernel, lengthscale 0.15). The posteriors interpolate smoothly between observations and honour each areal summary while preserving the short-lengthscale texture of the prior.

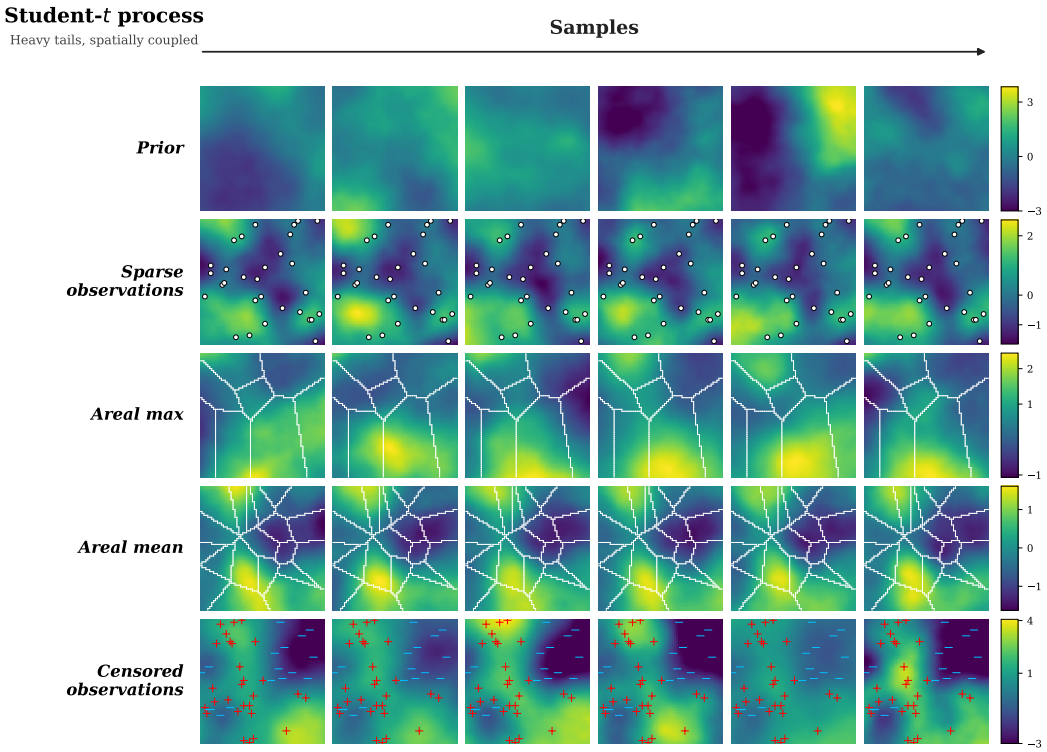


Figure 11: **Conditional sampling from a Student-t process prior.** The base process is a heavy-tailed, spatially coupled Student-t field ($\nu_t = 4$, Matérn-3/2 kernel, lengthscale 0.5).

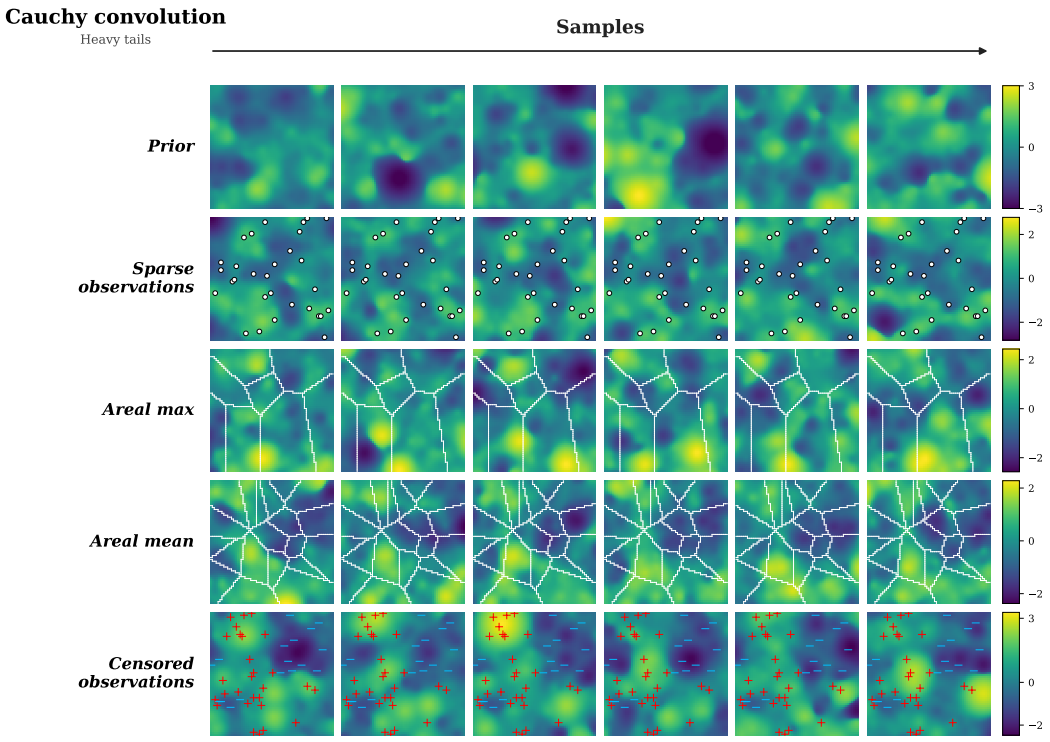


Figure 12: **Conditional sampling from a Cauchy convolution prior.** The base process is a very heavy-tailed field formed by convolving a Cauchy noise source with a short-lengthscale kernel (lengthscale 0.05). The infinite-variance source produces sharp, highly localised features; the posteriors reconcile these with the imposed point, areal, and censoring constraints.

G.3 Temporal Processes

We now turn our attention to the temporal processes considered in Sec. 5. We here consider a relaxation in which the terminal endpoint condition is approximated by a narrow Gaussian likelihood, rather than enforced exactly. Following Yang et al. (2025b), we consider four benchmarks. The Brownian motion and Ornstein–Uhlenbeck examples provide an analytically tractable Gaussian reference, while the cell-differentiation and FitzHugh–Nagumo examples test nonlinear and hypoelliptic settings.

G.3.1 Common Framework

Priors. Let $s \in [0, T]$ denote physical time, while $t \in [0, 1)$ denotes the latent noising time used by LatentFlow. We consider an additive-noise SDE given by

$$\mathrm{d}X_s = b_\vartheta(s, X_s)\,\mathrm{d}s + \Sigma_\vartheta(s, X_s)\,\mathrm{d}W_s.$$

We discretise this process on a physical-time grid $0 = s_0 < s_1 < \cdots < s_N = T$ where $\Delta s_n = s_{n+1} - s_n$ using an Euler–Maruyama discretisation, viz

$$X_{n+1} = X_n + b_\vartheta(s_n, X_n)\Delta s_n + \Sigma_\vartheta(s_n, X_n)\sqrt{\Delta s_n}\,\zeta_n, \tag{33}$$

where $\zeta_n \sim \mathcal{N}(0, I_{d_w})$ and d_w is the Brownian dimension. The latent innovation is thus the collection of standard Gaussian increments

$$\xi = (\zeta_0, \ldots, \zeta_{N-1}),$$

with $r(\xi) = \mathcal{N}(\xi; 0, I)$. Meanwhile, the process generator T_ϑ is the deterministic map obtained by iterating (33) from the fixed initial condition:

$$T_\vartheta(\xi) = (X_0, X_1, \ldots, X_N).$$

For brevity, model-specific formulas below suppress the fixed parameter ϑ; $h = \max_n \Delta s_n$ indexes the physical discretisation.

Condition. Given a discretised diffusion path $X = (X_0, \dots, X_N)$, with $X_n \approx X_{s_n}$, the endpoint likelihood is specified by

$$L_{\mathcal{C}}(X) = \mathcal{N}(y_{\mathcal{C}}; HX_N, \varepsilon_{\mathcal{C}}^2 I_{d_y}),$$

where H selects the observed coordinates and $\varepsilon_{\mathcal{C}}^2$ is the endpoint-observation variance. Equivalently, this corresponds to the observation model

$$y_{\mathcal{C}} = HX_N + \eta_{\mathcal{C}}, \qquad \eta_{\mathcal{C}} \sim \mathcal{N}(0, \varepsilon_{\mathcal{C}}^2 I_{d_y}).$$

Rare endpoint conditions are obtained by choosing $y_{\mathcal{C}}$ in the tail of the unconditioned terminal law. In our experiments, we set $\varepsilon_{\mathcal{C}} = 10^{-3}$ unless otherwise specified.

G.3.2 Evaluation

Ground truth We compare each sampler with a "ground truth" reference distribution. The primary references are analytic Gaussian laws for the linear models and independent exact rejection samples for every nonlinear model. All references use the same discretised process and finite-width Gaussian observation as the sampler. Specific details are provided for each model below.

Pathwise evaluation For pathwise comparisons, we evaluate each metric (see below) at the simulator nodes nearest the physical time points

$$\tau_j = T\{0.10 + 0.05(j-1)\}, \qquad j = 1, \dots, 17.$$

The endpoint is excluded from this evaluation grid and assessed separately. For a d-dimensional state, we stack all coordinates at these times into $p = 17d$ features. Let $\widehat{P}_j$ be the approximate marginal of feature j, and let $\widehat{Q}_j$ be the corresponding reference marginal. Let μ_j and ς_j denote the empirical reference mean and standard deviation, respectively. We report the mean standardised marginal Wasserstein distance, and the standardised mean and standard deviation errors, viz

$$\overline{W}_1 = \frac{1}{p}\sum_{j=1}^{p} \frac{W_1(\widehat{P}_j, \widehat{Q}_j)}{\varsigma_j}, \quad E_\mu = \left\{\frac{1}{p}\sum_{j=1}^{p}\left(\frac{\widehat{\mu}_j - \mu_j}{\varsigma_j}\right)^2\right\}^{1/2}, \quad E_{\mathrm{sd}} = \left\{\frac{1}{p}\sum_{j=1}^{p}\left(\frac{\widehat{\varsigma}_j}{\varsigma_j} - 1\right)^2\right\}^{1/2}.$$

Endpoint evaluation. For n paths and d_y observed coordinates, we evaluate the terminal endpoint condition using

$$\mathrm{RMSE}_{\mathrm{end}} = \left\{\frac{1}{nd_y}\sum_{i=1}^{n}\|HX_N^{(i)} - y_{\mathcal{C}}\|^2\right\}^{1/2}, \qquad \mathrm{Hit}_\delta = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left\{\frac{\|HX_N^{(i)} - y_{\mathcal{C}}\|}{\sqrt{d_y}} < \delta\right\}.$$

We use $\delta = 0.01$ for Brownian and OU examples, and $\delta = 0.05$ for cell and FHN examples. The endpoint bias and standard deviation are compared with their matched reference values, since a loose hit threshold can conceal excessive or insufficient endpoint dispersion.

Finite-sample reference scale. To account for natural sampling variability, we compare errors with those from equally sized batches generated from the reference distribution. For nonlinear models, these batches are resampled from a held-out reference pool; for linear models, they are drawn from the exact Gaussian law. We report the mean and central 95% range over 150 repetitions.

G.3.3 Brownian motion

Basic definition. Our first example is a Brownian motion with constant drift, defined according to

$$\mathrm{d}X_s = \gamma\,\mathrm{d}s + \sigma\,\mathrm{d}W_s$$

with $X_0 = 0, T = 1, y_{\mathcal{C}} = 0$. We vary the drift and diffusion scale over four different settings, shown in Tab. 5. The physical grid is uniform with $N = 500$ steps.

Table 5: **Model settings for Brownian motion**. The observation width is $\varepsilon_{\mathcal{C}} = 10^{-3}$. All terminal state coordinates are observed.

ID	Model	Parameters	x_0	$y_{\mathcal{C}}$	T	N
B1	Brownian	$(\gamma, \sigma) = (1, 1)$	0	0	1	500
B2	Brownian	$(\gamma, \sigma) = (1, 0.5)$	0	0	1	500
B3	Brownian	$(\gamma, \sigma) = (2, 1)$	0	0	1	500
B4	Brownian	$(\gamma, \sigma) = (-2, 1)$	0	0	1	500

Ground truth. For the Brownian motion and the Ornstein–Uhlenbeck (OU) process below, let $\mathbf{X}$ denote the vector of grid values. The prior law of this random vector is Gaussian with mean m and covariance K_h. For a scalar recursion $X_{k+1} = a_k X_k + c_k + q_k \zeta_k$, these quantities can be obtained via

$$m_{k+1} = a_k m_k + c_k, \qquad v_{k+1} = a_k^2 v_k + q_k^2, \qquad m_0 = x_0, \quad v_0 = 0,$$
$$(K_h)_{ij} = v_i \prod_{\ell=i}^{j-1} a_\ell, \qquad 0 \le i \le j \le N. \tag{34}$$

The Brownian motion has $(a_k, c_k, q_k) = (1, \gamma \Delta s_k, \sigma\sqrt{\Delta s_k})$; the OU process has $(a_k, c_k, q_k) = (1 - \gamma \Delta s_k, \gamma \mu \Delta s_k, \sigma\sqrt{\Delta s_k})$. If A_T selects the observed endpoint from $\mathbf{X}$, its exact conditional mean and covariance are

$$m_* = m + K_h A_T^\top (A_T K_h A_T^\top + \varepsilon_{\mathcal{C}}^2 I)^{-1} (y_{\mathcal{C}} - A_T m),$$
$$C_* = K_h - K_h A_T^\top (A_T K_h A_T^\top + \varepsilon_{\mathcal{C}}^2 I)^{-1} A_T K_h. \tag{35}$$

Sampling configuration. For each parameter configuration we use the LatentFlow SDE, with $S = 512$ Monte Carlo samples and $K = 1000$ integration steps.

Results. The results for the Brownian bridge are reported below. For evaluation, we use $R = 3$ independent batches of $n = 512$ paths per parameter configuration. Mean standardised marginal W_1 across seeds ranges from 0.047 to 0.070, while endpoint standard deviations are 0.976–1.020 times their exact conditional values. For pooled joint-interior means and endpoint variances, exact Gaussian chi-squared tests show no departures at the 5% level following Holm adjustment, separately for each diagnostic.

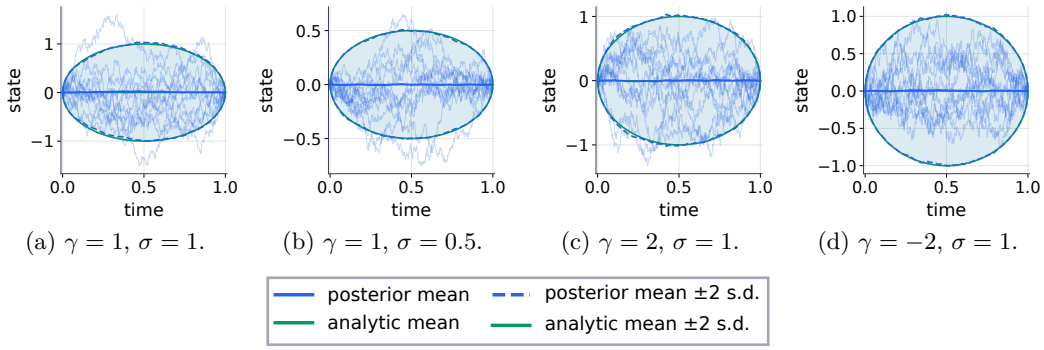
Table 6: **Pathwise evaluation for Brownian motion**. Smaller errors are better. Entries are mean ± sample SD across the stated independent sampler seeds. Floors are matched-size reference 2.5–97.5% ranges for one run.

ID	Method	$n \times R$	nW_1	Matched-n floor	Mean nRMSE	Cov. error
B1	LF	512×3	0.070 ± 0.015	[0.043, 0.085]	0.058 ± 0.021	0.084 ± 0.026
B2	LF	512×3	0.047 ± 0.002	[0.042, 0.083]	0.026 ± 0.003	0.081 ± 0.033
B3	LF	512×3	0.052 ± 0.007	[0.042, 0.084]	0.034 ± 0.007	0.089 ± 0.017
B4	LF	512×3	0.053 ± 0.009	[0.042, 0.086]	0.042 ± 0.014	0.077 ± 0.012

Table 7: **Endpoint evaluation for Brownian motion**. Endpoint RMSE is first pooled over paths within each run; entries then give mean ± SD across seeds. Endpoint SD/$\varepsilon_{\mathcal{C}}$ averages observed-coordinate sample SD divided by the likelihood width.

ID	Method	Endpoint RMSE	Endpoint SD/$\varepsilon_{\mathcal{C}}$	Hit (%)	Misses/total	Nonfinite
B1	LF	0.00099 ± 0.00002	0.995 ± 0.020	100.0 ± 0.0	0/1536	0
B2	LF	0.00100 ± 0.00005	0.995 ± 0.045	100.0 ± 0.0	0/1536	0
B3	LF	0.00098 ± 0.00004	0.981 ± 0.044	100.0 ± 0.0	0/1536	0
B4	LF	0.00098 ± 0.00004	0.975 ± 0.043	100.0 ± 0.0	0/1536	0

The Brownian sample paths, means and pointwise standard deviations appear in Fig. 13.



(a) $\gamma = 1$, $\sigma = 1$. (b) $\gamma = 1$, $\sigma = 0.5$. (c) $\gamma = 2$, $\sigma = 1$. (d) $\gamma = -2$, $\sigma = 1$.

Figure 13: **Conditional samples for the Brownian motion.** Thin blue curves show sample LatentFlow paths; solid blue curves show means over all LatentFlow paths. The four panels vary the constant drift γ and the diffusion scale σ. Blue dashed bounds are LatentFlow means ± 2 pointwise standard deviations; green solid means and bounds are exact conditional Gaussian moments for the same grid and observation width.

G.3.4 Ornstein–Uhlenbeck diffusion

Basic definition. We next consider a standard Ornstein–Uhlenbeck (OU) process, given by

$$\mathrm{d}X_s = \gamma(\mu - X_s)\,\mathrm{d}s + \sigma\,\mathrm{d}W_s, \qquad X_0 = 0, \quad T = 1,$$

on the same 500-step grid. The settings vary the mean-reversion rate, equilibrium mean, diffusion scale and terminal target.

Table 8: **Model settings for OU process**. The observation width is $\varepsilon_{\mathcal{C}} = 10^{-3}$. All terminal state coordinates are observed.

ID	Model	Parameters	x_0	yc	T	N
O1	OU	$(\gamma, \mu, \sigma) = (1.7, 1, 0.3)$	0	1	1	500
O2	OU	$(\gamma, \mu, \sigma) = (0.6, 1, 0.3)$	0	1	1	500
O3	OU	$(\gamma, \mu, \sigma) = (1.7, 1, 1)$	0	1	1	500
O4	OU	$(\gamma, \mu, \sigma) = (1.7, 0, 0.3)$	0	0	1	500

Ground truth. The exact Gaussian reference can be computed using (34)–(35), with $a_k = 1 - \gamma\Delta s_k$, $c_k = \gamma\mu\Delta s_k$ and $q_k = \sigma\sqrt{\Delta s_k}$.

Sampling configuration. For each parameter configuration we use the LatentFlow SDE, with $S = 512$ Monte Carlo samples and $K = 1000$ integration steps.

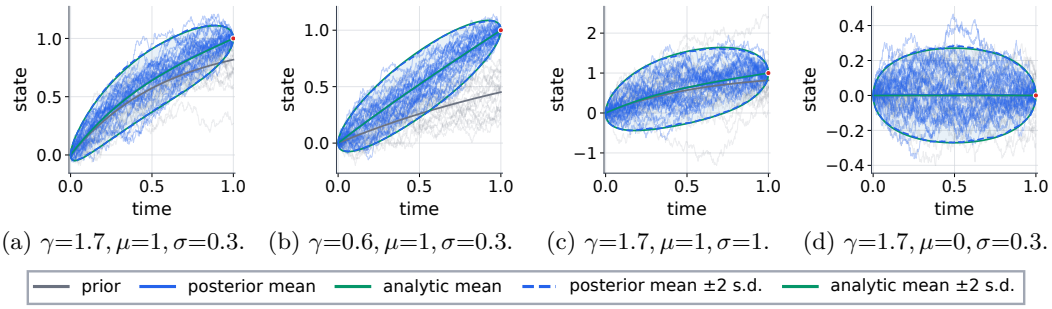
Results. Tabs. 9 and 10 give the pathwise and endpoint results for the OU bridge. Once more, we use $R = 3$ independent batches of $n = 512$ paths per parameter configuration. Meanwhile, Fig. 14 shows sample paths together with the LF and analytic means and pointwise standard deviations.

Table 9: **Pathwise evaluation for OU process**. Smaller errors are better. Entries are mean ± sample SD across the stated independent sampler seeds. Floors are matched-size reference 2.5–97.5% ranges for one run.

ID	Method	$n \times R$	nW_1	Matched-n floor	Mean nRMSE	Cov. error
O1	LF	512×3	0.053 ± 0.008	[0.043, 0.081]	0.036 ± 0.019	0.103 ± 0.025
O2	LF	512×3	0.052 ± 0.008	[0.042, 0.084]	0.036 ± 0.011	0.091 ± 0.059
O3	LF	512×3	0.054 ± 0.007	[0.043, 0.082]	0.033 ± 0.016	0.102 ± 0.024
O4	LF	512×3	0.060 ± 0.013	[0.043, 0.083]	0.047 ± 0.021	0.096 ± 0.029

Table 10: **Endpoint evaluation for OU process**. Endpoint RMSE is first pooled over paths within each run; entries then give mean ± SD across seeds. Endpoint SD/$\varepsilon_{\mathcal{C}}$ averages observed-coordinate sample SD divided by the likelihood width.

ID	Method	Endpoint RMSE	Endpoint SD/$\varepsilon_{\mathcal{C}}$	Hit (%)	Misses/total	Nonfinite
O1	LF	0.00101 ± 0.00002	1.013 ± 0.025	100.0 ± 0.0	0/1536	0
O2	LF	0.00102 ± 0.00006	1.020 ± 0.059	100.0 ± 0.0	0/1536	0
O3	LF	0.00101 ± 0.00004	1.012 ± 0.036	100.0 ± 0.0	0/1536	0
O4	LF	0.00098 ± 0.00004	0.980 ± 0.038	100.0 ± 0.0	0/1536	0



(a) γ=1.7, μ=1, σ=0.3. (b) γ=0.6, μ=1, σ=0.3. (c) γ=1.7, μ=1, σ=1. (d) γ=1.7, μ=0, σ=0.3.

Figure 14: **Conditional samples for the OU process.** The four panels vary the mean-reversion rate γ, equilibrium mean μ, and diffusion scale σ. Grey trajectories are prior draws; blue trajectories and solid curves are LatentFlow draws and their empirical means. Solid green curves are analytic conditional means for the same Euler grid and Gaussian observation width $\varepsilon_{\mathcal{C}} = 10^{-3}$. Red markers show the target endpoints. Blue dashed and green solid bounds show the LatentFlow and analytic means ±2 pointwise standard deviations, respectively. Light shading fills the bounds.

G.3.5 Cell-differentiation diffusion

Basic definition. We consider the two-dimensional cell-differentiation model from Wang et al. (2011); Baker et al. (2025)

$$\mathrm{d}X_s = \begin{pmatrix} 1 + u(X_s^1) - u(X_s^2) - X_s^1 \\ 1 + u(X_s^2) - u(X_s^1) - X_s^2 \end{pmatrix} \mathrm{d}s + \sigma I_2 \, \mathrm{d}W_s, \qquad u(a) = \frac{a^4}{2^{-4} + a^4},$$

with $\sigma = 0.1$ and $N = 400$ uniform time steps. We observe both terminal coordinates and consider normal, rare and symmetric endpoint conditions, as specified in Tab. 11.

Table 11: **Model settings for cell differentiation**. The observation width is $\varepsilon_{\mathcal{C}} = 10^{-3}$. All terminal state coordinates are observed.

ID	Model	Parameters	x_0	$y_{\mathcal{C}}$	T	N
C1	Cell, normal	$\sigma = 0.1$	$(0.1, -0.1)$	$(2, -0.1)$	4	400
C2	Cell, rare	$\sigma = 0.1$	$(0.1, -0.1)$	$(1, -0.1)$	4	400
C3	Cell, symmetric	$\sigma = 0.1$	$(-1, -1)$	$(-1, -1)$	5	400

Ground truth. We construct independent reference paths via rejection sampling. In each case, we first sample the path up to the penultimate time step, $(X_0, \ldots, X_{N-1})$, and then sample the endpoint conditionally.

For the normal condition, we generate this partial path from the unconditioned Euler scheme. Given its final state $x = X_{N-1}$, the remaining Euler step has distribution

$$X_N \mid x \sim \mathcal{N}(m_x, vI_2), \qquad m_x = x + \Delta s_{N-1} b(x), \qquad v = \sigma^2 \Delta s_{N-1}.$$

Combining this transition with the observation noise gives $y_{\mathcal{C}} \mid x \sim \mathcal{N}(m_x, (v + \varepsilon_{\mathcal{C}}^2) I_2)$. We therefore accept the partial path with probability

$$\exp\left\{ -\frac{\|m_x - y_{\mathcal{C}}\|^2}{2(v + \varepsilon_{\mathcal{C}}^2)} \right\}.$$

Given an accepted path, we obtain a sample from the full trajectory by drawing the endpoint according to

$$X_N \mid x, y_{\mathcal{C}} \sim \mathcal{N}\left(\frac{\varepsilon_{\mathcal{C}}^2 m_x + v y_{\mathcal{C}}}{v + \varepsilon_{\mathcal{C}}^2}, \frac{v\varepsilon_{\mathcal{C}}^2}{v + \varepsilon_{\mathcal{C}}^2} I_2\right). \tag{36}$$

This produces independent samples from the finite-grid conditional law. By averaging over the final process noise before rejection, we avoid having to simulate an endpoint that happens to fall within the narrow observation likelihood.

For the rare and symmetric conditions, rejection from the unconditioned process remains infeasible; see below for further details. We instead first account for the observed terminal sum analytically. Define the sum and difference coordinates

$$U_s = \frac{X_s^1 + X_s^2}{\sqrt{2}}, \qquad D_s = \frac{X_s^1 - X_s^2}{\sqrt{2}},$$

and define $y_{\mathcal{C},U}$ and $y_{\mathcal{C},D}$ by the same transformation of $y_{\mathcal{C}}$. The sum evolves independently of the difference:

$$\mathrm{d}U_s = (\sqrt{2} - U_s)\,\mathrm{d}s + \sigma\,\mathrm{d}B_s.$$

Its Euler discretisation is therefore linear and Gaussian. We can consequently draw the noise increments driving U up to time s_{N-1} exactly conditional on $y_{\mathcal{C},U}$, accounting for both the remaining Euler step and the observation noise. We independently draw the noise increments driving D from their original standard Gaussian law. Together, these increments, denoted by ξ_-, have a Gaussian distribution Q and generate a partial path through the Euler scheme. This proposal accounts for the sum observation; thus only the difference observation remains to be imposed. We accept each proposed partial path with probability

$$L_D(\xi_-) = \exp\left\{-\frac{(m_{N,D}(\xi_-) - y_{\mathcal{C},D})^2}{2(v + \varepsilon_{\mathcal{C}}^2)}\right\},$$

where $m_{N,D}(\xi_-)$ is the difference coordinate of $m_x = x + \Delta s_{N-1} b(x)$ for the proposed state $x = X_{N-1}$. Finally, we draw X_N in the original coordinates using (36). The resulting draws are independent samples from the same finite-grid conditional law: the sum observation is imposed analytically, and the difference observation is imposed by rejection.

Rarity of the endpoint conditions. For C2 and C3, the observed sum $y_{\mathcal{C},U}$ lies approximately 10.6 and 39.6 prior standard deviations from its mean, respectively. Indeed, writing $U_N \sim \mathcal{N}(m_U, v_U)$ for the Euler prior, direct rejection with acceptance probability $\exp\{-\|X_N - y_{\mathcal{C}}\|^2/(2\varepsilon_{\mathcal{C}}^2)\}$ has mean acceptance probability bounded by

$$p_{\mathrm{acc}} \le \frac{\varepsilon_{\mathcal{C}}}{\sqrt{v_U + \varepsilon_{\mathcal{C}}^2}} \exp\left\{-\frac{(y_{\mathcal{C},U} - m_U)^2}{2(v_U + \varepsilon_{\mathcal{C}}^2)}\right\}.$$

On our grids, these upper bounds are below 5×10^{-27} and 3×10^{-343}, respectively, making direct prior rejection computationally infeasible. Consistently, Yang et al. (2025b, Appendix B.3) report no valid paths in 100,000 forward simulations for either condition, using an endpoint-ball tolerance of 0.01. Our independent references remain feasible because analytic sum conditioning removes this rare Gaussian factor before rejection against the residual difference likelihood.

Sampling configuration. For C1, we use the standard LatentFlow SDE with $S = 128$ Monte Carlo samples and $K = 4000$ integration steps. For C2 and C3, we use an additional analytic conditioning step before applying LatentFlow: the known sum observation is absorbed into the Gaussian base Q, and the final innovation is integrated out. We then apply ordinary LatentFlow to whitened coordinates of Q, with $S = 128$, $K = 1000$ and residual likelihood L_D, before drawing the endpoint conditionally. In all three cases, we generate $R = 3$ independent batches of $n = 128$ paths. For C2, we additionally report a pCN correction alongside LatentFlow.

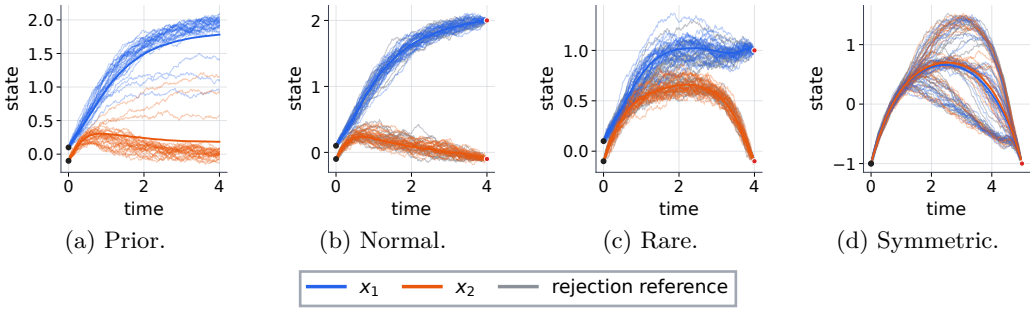
Results. Tabs. 12 and 13 report pathwise and endpoint accuracy. Meanwhile, samples from the posterior are shown in Fig. 15, and posterior marginals are shown in Fig. 16. For C1, standard LF gives pooled $\overline{W}_1 = 0.0843$, within the matched-size reference range $[0.0698, 0.1039]$. For C3, LF gives $\overline{W}_1 = 0.0865$ and captures both modes. For C2, LF gives $\overline{W}_1 = 0.301$ and covariance error 2.125; pCN reduces these to 0.0777 and 0.169. Consistent with these metrics, the samples generated by LatentFlow are qualitatively very closely aligned with those from the ground truth reference.

Table 12: **Pathwise evaluation for cell differentiation**. Smaller errors are better. Entries are mean ± sample SD across the stated independent sampler seeds. Floors are matched-size reference 2.5–97.5% ranges for one run.

ID	Method	$n \times R$	nW_1	Matched-n floor	Mean nRMSE	Cov. error
C1	LF	128×3	0.123 ± 0.017	[0.100, 0.159]	0.106 ± 0.019	0.213 ± 0.015
C2	LF	128×3	0.315 ± 0.084	[0.096, 0.173]	0.323 ± 0.072	2.170 ± 1.748
C2	LF+pCN	128×3	0.107 ± 0.005	[0.096, 0.173]	0.074 ± 0.014	0.238 ± 0.035
C3	LF	128×3	0.130 ± 0.036	[0.064, 0.178]	0.111 ± 0.039	0.179 ± 0.073

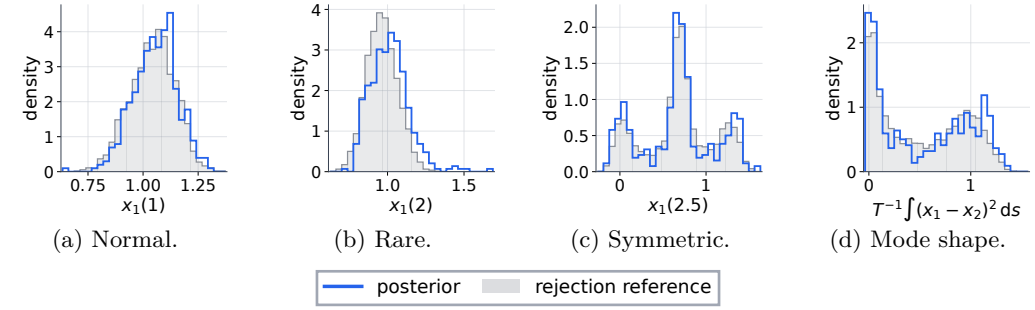
Table 13: **Endpoint evaluation for cell differentiation.** Endpoint RMSE is first pooled over paths within each run; entries then give mean ± SD across seeds. Endpoint SD/$\varepsilon_{\mathcal{C}}$ averages observed-coordinate sample SD divided by the likelihood width.

ID	Method	Endpoint RMSE	Endpoint SD/$\varepsilon_{\mathcal{C}}$	Hit (%)	Misses/total	Nonfinite
C1	LF	0.00114 ± 0.00006	1.100 ± 0.061	100.0 ± 0.0	0/384	0
C2	LF	0.00102 ± 0.00003	1.019 ± 0.039	100.0 ± 0.0	0/384	0
C2	LF+pCN	0.00098 ± 0.00005	0.978 ± 0.054	100.0 ± 0.0	0/384	0
C3	LF	0.00107 ± 0.00002	1.013 ± 0.017	100.0 ± 0.0	0/384	0



(a) Prior. (b) Normal. (c) Rare. (d) Symmetric.

Figure 15: **Cell-differentiation bridges.** Panels show the prior and LatentFlow paths for the normal, rare and symmetric endpoint conditions. Blue and orange denote the two state coordinates; light grey paths in the posterior panels are independent references. Solid curves are sample means. Black and red markers show initial states and endpoint observations.



(a) Normal. (b) Rare. (c) Symmetric. (d) Mode shape.

Figure 16: **Cell-differentiation marginals.** LatentFlow samples are compared with independent references at the indicated times. The coordinate-separation functional additionally checks within-mode geometry for C3.

G.3.6 FitzHugh–Nagumo diffusion

Definition. Finally, we consider a FitzHugh–Nagumo model (e.g., Bierkens et al., 2020; Yang et al., 2025b) given by

$$\begin{aligned} \mathrm{d}V_s &= \chi^{-1}(V_s - R_s - V_s^3 + I_{\mathrm{ext}})\,\mathrm{d}s, \\ \mathrm{d}R_s &= (\gamma V_s - R_s + \alpha)\,\mathrm{d}s + \sigma\,\mathrm{d}W_s, \end{aligned} \qquad (\chi, I_{\mathrm{ext}}, \gamma, \alpha, \sigma) = (0.1, 0, 1.5, 0.8, 0.3).$$

We take $X_0 = (-0.5, -0.6)$ and $T = 2$, and observe only V_T, near either -1 or 1.1. The recovery endpoint is unobserved. Noise acts directly only on recovery, making this a hypoelliptic example. We use $\varepsilon_{\mathcal{C}} = 10^{-2}$ and $N = 200$ time steps, concentrated near the endpoint:

$$u_k = \frac{kT}{N}, \qquad s_k = u_k\left(2 - \frac{u_k}{T}\right), \qquad k = 0, \ldots, N.$$

Table 14: **Model settings for FitzHugh–Nagumo diffusion**. The observation width is $\varepsilon_{\mathcal{C}} = 10^{-2}$. Only terminal voltage is observed.

ID	Model	Parameters	x_0	$y_{\mathcal{C}}$	T	N
F1	FHN, low	(G.3.6)	$(-0.5, -0.6)$	-1	2	200
F2	FHN, high	(G.3.6)	$(-0.5, -0.6)$	1.1	2	200

Ground truth. We generate reference samples via rejection sampling. In particular, we draw independent prior Euler–Maruyama paths and accept each with probability

$$\exp\left\{-\frac{(V_N - y_{\mathcal{C}})^2}{2\varepsilon_{\mathcal{C}}^2}\right\}.$$

This gives independent samples from the same posterior targeted by LatentFlow.

Sampling configuration. Both conditions use the standard LatentFlow SDE with $K = 2000$ integration steps. F1 uses $S = 128$ Monte Carlo samples and $R = 3$ batches of $n = 128$ paths; F2 uses $S = 16384$ and $R = 3$ batches of $n = 64$ paths. For F2, we also report an innovation-space pCN correction alongside standard LF.

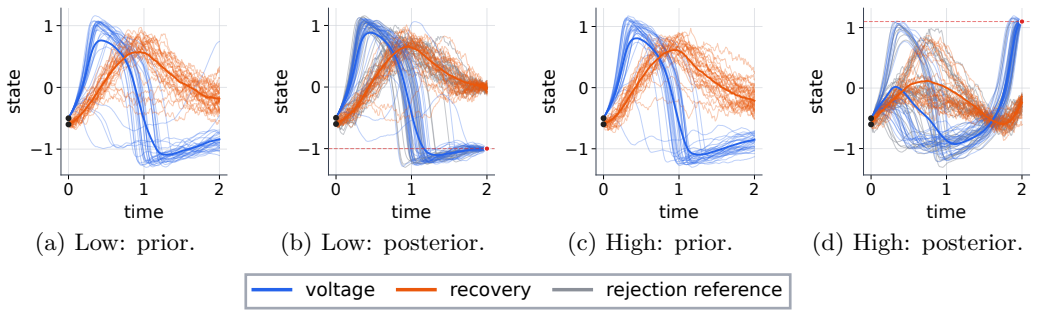
Results. The pathwise and endpoint results appear in Tabs. 15 and 16. Meanwhile, samples from the posterior are shown in Fig. 17, and posterior marginals are shown in Fig. 18. For F1, pooled $\overline{W}_1 = 0.0823$ lies within the reference range $[0.0542, 0.0937]$, and endpoint RMSE is 0.00981, compared with 0.01006 for the reference. For F2, LatentFlow generates qualitatively convincing sample paths. Nonetheless, the pooled $\overline{W}_1 = 0.2360$ and covariance error 0.6020 are marginally above the reference ranges $[0.0701, 0.1487]$ and $[0.1029, 0.2819]$. The pCN correction reduces these to 0.0775 and 0.1637, respectively.

Table 15: **Pathwise evaluation for FitzHugh–Nagumo diffusion**. Smaller errors are better. Entries are mean ± sample SD across the stated independent sampler seeds. Floors are matched-size reference 2.5–97.5% ranges for one run.

ID	Method	$n \times R$	nW_1	Matched-n floor	Mean nRMSE	Cov. error
F1	LF	128×3	0.124 ± 0.032	$[0.088, 0.156]$	0.094 ± 0.049	1.071 ± 0.668
F2	LF	64×3	0.264 ± 0.054	$[0.112, 0.236]$	0.241 ± 0.060	0.643 ± 0.040
F2	LF+pCN	64×3	0.136 ± 0.029	$[0.112, 0.236]$	0.096 ± 0.043	0.249 ± 0.095

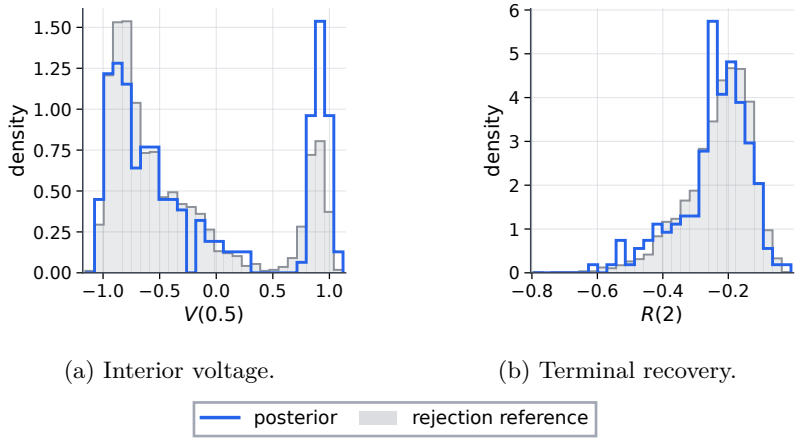
Table 16: **Endpoint evaluation for FitzHugh–Nagumo diffusion**. Endpoint RMSE is first pooled over paths within each run; entries then give mean ± SD across seeds. Endpoint SD/$\varepsilon_{\mathcal{C}}$ averages observed-coordinate sample SD divided by the likelihood width.

ID	Method	Endpoint RMSE	Endpoint SD/$\varepsilon_{\mathcal{C}}$	Hit (%)	Misses/total	Nonfinite
F1	LF	0.00980 ± 0.00037	0.983 ± 0.038	100.0 ± 0.0	0/384	0
F2	LF	0.00895 ± 0.00061	0.886 ± 0.048	100.0 ± 0.0	0/192	0
F2	LF+pCN	0.01067 ± 0.00093	1.051 ± 0.085	100.0 ± 0.0	0/192	0



(a) Low: prior. (b) Low: posterior. (c) High: prior. (d) High: posterior.

Figure 17: **Posterior samples for the FitzHugh–Nagumo process**. Panels show prior and uncorrected LatentFlow paths for the low and high voltage observations. Blue and orange denote voltage and recovery; light grey paths in the posterior panels are independent references. Solid curves are sample means. Black and red markers show initial states and observed endpoints. Only voltage is conditioned.



(a) Interior voltage. (b) Terminal recovery.

Figure 18: **Posterior marginals for the FitzHugh–Nagumo process.** Uncorrected LatentFlow samples for the high-endpoint condition are compared with independent references for interior voltage and unobserved terminal recovery.

G.4 Spatio-temporal Processes

We next provide additional numerical results for the spatiotemporal examples shown in Fig. 5, as well as full implementation details.

Each model is implemented as a deterministic map from a Gaussian latent $\mathbf{z}$ to a spatio-temporal field trajectory, enabling the use of LatentFlow. All posteriors are obtained by running the integrator sampler for 200 steps with an Euler Maruyama SDE integration and a single Monte Carlo guidance sample per step.

G.5 Advection-diffusion plume

G.5.1 Forward Model

The simulator models a continuously emitting point source on a 22×22 spatial grid with domain $[-1, 1]^2$ over $T = 32$ time steps at horizon $t_{\max} = 2.2$. The full latent vector $\mathbf{z}$ has the following components, each drawn independently from $\mathcal{N}(0, 1)$:

- $z_x^{(s)}$, $z_y^{(s)}$: control the source location (x_s, y_s) via a tanh squashing, confining the emitter to a small region near $(-0.76, 0)$;
- $z^{(w)}$: sets the mean horizontal wind speed $\bar{u}_x$;
- $z^{(y)}$: sets the base lateral (cross-wind) drift u_y^{base};
- $z_0^{(I)}, z_1^{(I)}, \ldots, z_{T-1}^{(I)}$: a time-indexed sequence controlling the emission intensity at each step through a cumulative tanh-squashed process $\xi^{(I)}$;
- $z_0^{(w)}, \ldots, z_{T-1}^{(w)}$: a time-indexed sequence driving the fluctuating component of the horizontal wind $\xi^{(w)}$;
- two sets of i.i.d. latents driving the slow and fast lateral meander processes ξ^{slow} and ξ^{fast}.

LatentFlow performs inference over this entire latent vector, recovering posteriors over source location, wind field, and emission history jointly from observations of the concentration field. The source location (x_s, y_s) is stochastic, with

$$x_s = -0.76 + 0.12 \tanh(z_x^{(s)}), \qquad y_s = 0.45 \tanh(z_y^{(s)}).$$

The concentration field at time t_k is the superposition of all past Gaussian parcels,

$$c(\mathbf{x}, t_k) = \sum_{j=0}^{k} \frac{\Delta t \cdot \alpha_j \cdot I_j}{2\pi\sigma_j^2} \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_{kj}\|^2}{2\sigma_j^2}\right),$$

where $\sigma_j^2 = \sigma_0^2 + 2\kappa(t_k - t_j)$ with $\sigma_0 = 0.040$ and diffusivity $\kappa = 0.028$; $\alpha_j = e^{-0.28(t_k - t_j)}$ is an age-decay factor; and I_j is a stochastic intensity,

$$\log I_j = 0.28\, \xi_j^{(I)} + 0.18\, z_0^{(I)}, \qquad \xi_j^{(I)} = \tanh\left(0.38 \sum_{i=0}^{j} z_i^{(I)} \sqrt{\Delta t}\right).$$

The parcel centre for a parcel emitted at t_j, evaluated at t_k, is

$$\mu_{kj}^x = x_s + \bar{u}_x^j (t_k - t_j), \qquad \mu_{kj}^y = y_s + u_y^{\text{base}}(t_k - t_j) + 0.52\, \Delta\xi_{kj}^{\text{slow}} + 0.18\, \Delta\xi_{kj}^{\text{fast}},$$

where $\bar{u}_x^j = 0.72 + 0.08 \tanh(z^{(w)}) + 0.06 \xi_j^{(w)}$ is the horizontal wind, $u_y^{\text{base}} = 0.30 \tanh(z^{(y)})$ is a stochastic base lateral drift, and $\Delta\xi_{kj}^{\text{slow}}$, $\Delta\xi_{kj}^{\text{fast}}$ are increments of two independent tanh-squashed cumulative Brownian meander processes.

G.5.2 Conditions

We consider three different conditioning regimes (Fig. 19). One of these is shown in the main text (Fig. 5).

Sparse sensors. A ground-truth trajectory is sampled from the prior. Eight sensor locations are selected as the top-8 highest-concentration pixels in the right half of the domain ($x > 0.20$), averaged over four observation times $t \in \{t_8, t_{14}, t_{20}, t_{26}\}$, giving $8 \times 4 = 32$ scalar observations in total. For each sensor-time pair (i, j), the condition is an equality constraint,

$$c(\mathbf{x}_i, t_j) = c^*(\mathbf{x}_i, t_j) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2),\ \sigma = 0.02,$$

Obstacle avoidance. The plume must not enter a protected central rectangular region $\mathcal{P} = \{(x, y) : 0.32 < x < 0.82,\ |y| < 0.34\}$ at any time step. This is a binary condition with smooth margin

$$\phi(c) = 0.06 - \text{smax}_t\big(\text{smax}_{\mathbf{x} \in \mathcal{P}}\big[c(\mathbf{x}, t)\big]\big) > 0,$$

where smax denotes the log-sum-exp smooth maximum with temperature $\tau = 0.025$.

Detector hit. A small downstream detector region $\mathcal{D} = \{(x, y) : 0.55 < x < 0.90,\ y < -0.45\}$ must be triggered, i.e., the peak concentration over the detector must exceed a threshold $c^* = 0.40$:

$$\phi(c) = \mathrm{smax}_t\big(\mathrm{smax}_{\mathbf{x}\in\mathcal{D}}\big[c(\mathbf{x}, t)\big]\big) - 0.40 > 0.$$

This event is rare under the prior, so guidance must steer samples toward trajectories in which the plume reaches the bottom-right corner.

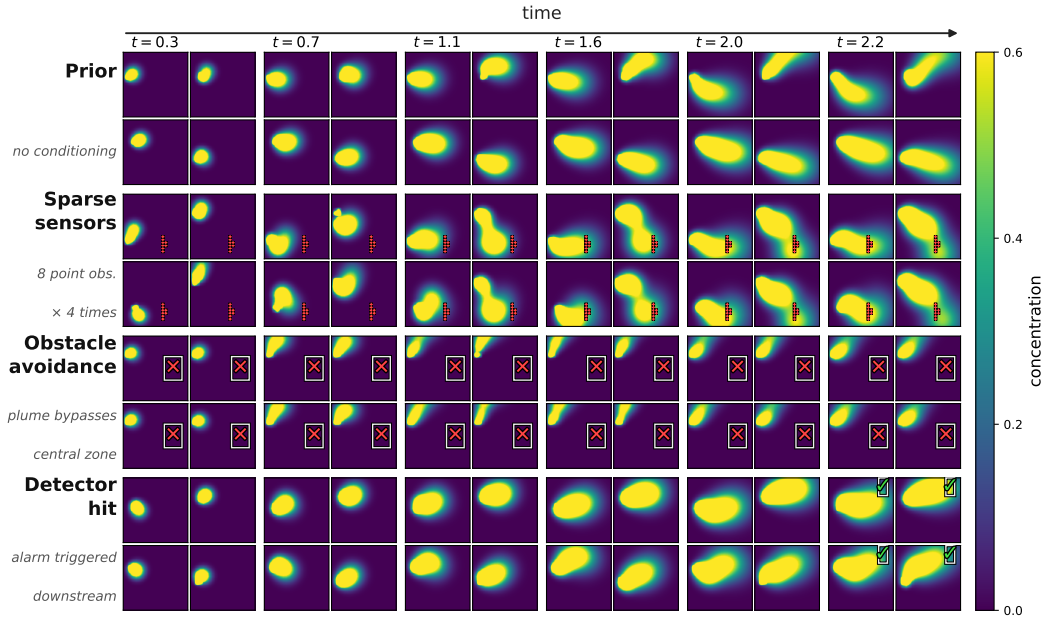


Figure 19: **Posterior samples from the advection–diffusion plume model under three conditioning regimes, each showing four independent draws at six time snapshots.** *Prior:* unconditioned samples exhibiting diverse plume trajectories and intensities. *Sparse sensors:* conditioning on exact concentration values at 8 spatial locations observed at 4 times (32 observations total) recovers the ground-truth plume trajectory. *Obstacle avoidance:* a binary condition requiring the plume to stay below concentration 0.06 in the central rectangular region (dashed box, ×) at all times steers samples around the obstacle. *Detector hit:* a binary condition requiring peak concentration to exceed 0.40 in the downstream detector region (dashed box, ✓) concentrates mass on trajectories that reach the top-right corner.

G.6 FitzHugh–Nagumo excitable medium

G.6.1 Forward Model

The simulator integrates the FitzHugh–Nagumo reaction-diffusion system on a 20×20 periodic grid over $T = 34$ steps at horizon $t_{\max} = 4.0$.

The latent vector $\mathbf{z}$ encodes the following independent $\mathcal{N}(0, 1)$ draws:

- $z_t^{(1)}$: shifts the firing time of pacemaker 1 (upper-left) within the window $[0.27, 0.83]$ via $t_1 = 0.55 + 0.28\tanh(z_t^{(1)})$;
- $z_t^{(2)}$: shifts the firing time of pacemaker 2 (lower-right) within the window $[1.70, 2.90]$ via $t_2 = 2.30 + 0.60\tanh(z_t^{(2)})$;
- two additional scalars controlling the amplitudes A_1 and A_2 of each pacemaker;
- $z^{(\eta)}$: a single scalar that, after box-filter smoothing, produces the fixed spatial heterogeneity field $\eta(\mathbf{x})$ modulating local excitability across the grid.

The SDE noise terms $\mathrm{d}W_u$ and $\mathrm{d}W_v$ in (37)–(38) are not part of $\mathbf{z}$; they are injected directly during simulation and are not inferred. LatentFlow thus performs inference over the low-dimensional latent $\mathbf{z}$ controlling which pacemakers fire, when, and how strongly.

The activator u and recovery variable v evolve as

$$\mathrm{d}u = \Big[D_u \nabla^2 u + u - \tfrac{u^3}{3} - v + I_{\mathrm{stim}}(\mathbf{x}, t) + \eta(\mathbf{x})\Big]\,\mathrm{d}t + \sigma_u\,\mathrm{d}W_u, \tag{37}$$

$$\mathrm{d}v = \epsilon\,(u + a - bv)\,\mathrm{d}t + \sigma_v\,\mathrm{d}W_v, \tag{38}$$

with $D_u = 0.20$, $\epsilon = 0.095$, $a = 0.72$, $b = 0.82$, $\sigma_u = 0.12$, $\sigma_v = 0.035$. After each Euler–Maruyama step both variables are saturated via $u \leftarrow 2.25\tanh(u/2.25)$, $v \leftarrow 1.65\tanh(v/1.65)$. The heterogeneity field $\eta(\mathbf{x}) = 0.11\,\mathcal{S}^{(4)}(z^{(\eta)})$ is a spatially smooth noise field constructed by applying four passes of a box-filter smoother $\mathcal{S}$ to the global latent channel and held fixed over time.

Two spatially localised Gaussian pacemakers provide the stimulus:

$$I_{\mathrm{stim}}(\mathbf{x}, t) = \sum_{i=1}^{2} A_i\, \exp\!\left(-\frac{(t - t_i)^2}{2 \cdot 0.30^2}\right) \exp\!\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2 \cdot 0.17^2}\right),$$

with source 1 at $\mathbf{x}_1 \approx (-0.72,\, 0.55)$ (upper-left) and firing time $t_1 = 0.55 + 0.28\tanh(z_t^{(1)}) \in [0.27, 0.83]$, and source 2 at $\mathbf{x}_2 \approx (0.68,\, -0.55)$ (lower-right) and $t_2 = 2.30 + 0.60\tanh(z_t^{(2)}) \in [1.70, 2.90]$. The amplitudes A_i are also drawn stochastically.

G.6.2 Conditions

We consider three binary conditioning regimes. Let $p(\mathbf{x}, t; \mathcal{M}, \mathcal{T})$ denote the peak activator value over spatial region $\mathcal{M}$ and time window $\mathcal{T}$:

$$p(u; \mathcal{M}, \mathcal{T}) = \max_{t \in \mathcal{T}} \max_{\mathbf{x} \in \mathcal{M}} u(\mathbf{x}, t).$$

Define neighbourhood regions $\mathcal{N}_1 = \{x < -0.45,\, y > 0.30\}$ and $\mathcal{N}_2 = \{x > 0.45,\, y < -0.30\}$ for the two sources, and time windows $\mathcal{T}_{\mathrm{early}} = [0, 1.30]$ and $\mathcal{T}_{\mathrm{late}} = [1.80, 4.0]$.

Source 1 only. The lower-left pacemaker fires strongly early while the upper-right stays silent:

$$p(u; \mathcal{N}_1, \mathcal{T}_{\mathrm{early}}) > 0.55 \quad \text{and} \quad p(u; \mathcal{N}_2, \mathcal{T}_{\mathrm{late}}) < -0.75.$$

Source 2 only. The upper-right pacemaker fires late while the lower-left never reaches threshold:

$$p(u; \mathcal{N}_2, \mathcal{T}_{\mathrm{late}}) > 0.55 \quad \text{and} \quad p(u; \mathcal{N}_1, \mathcal{T}_{\mathrm{early}}) < -0.75.$$

Both fire. Sequential activation: lower-left fires early, upper-right fires late:

$$p(u; \mathcal{N}_1, \mathcal{T}_{\mathrm{early}}) > 0.55 \quad \text{and} \quad p(u; \mathcal{N}_2, \mathcal{T}_{\mathrm{late}}) > 0.65.$$

All three are implemented as binary conditions with probit likelihood and $\nu = 10^{-5}$ (10^{-3} for source 2 only), with 4 guidance samples per step and gradient norm clipping at 30.0.

G.7 Allen–Cahn phase field

G.7.1 Forward Model

The simulator integrates the stochastic Allen–Cahn equation on a 64×64 periodic grid over $T = 30$ steps with $\Delta t = 1/29$:

$$\mathrm{d}u = \big[D\nabla^2 u + r\,(u - u^3)\big]\,\mathrm{d}t + \sigma\,\mathrm{d}W,$$

with $D = 2.0$, $r = 1.45$, $\sigma = 0.45$. The double-well potential $f(u) = u - u^3 = -\nabla_u F(u)$ for $F(u) = \frac{1}{4}(1 - u^2)^2$ drives u toward ± 1, with characteristic interface width $\ell \sim \sqrt{D/r} \approx 1.17$ grid units. The Laplacian uses a nearest-neighbour finite-difference stencil with periodic

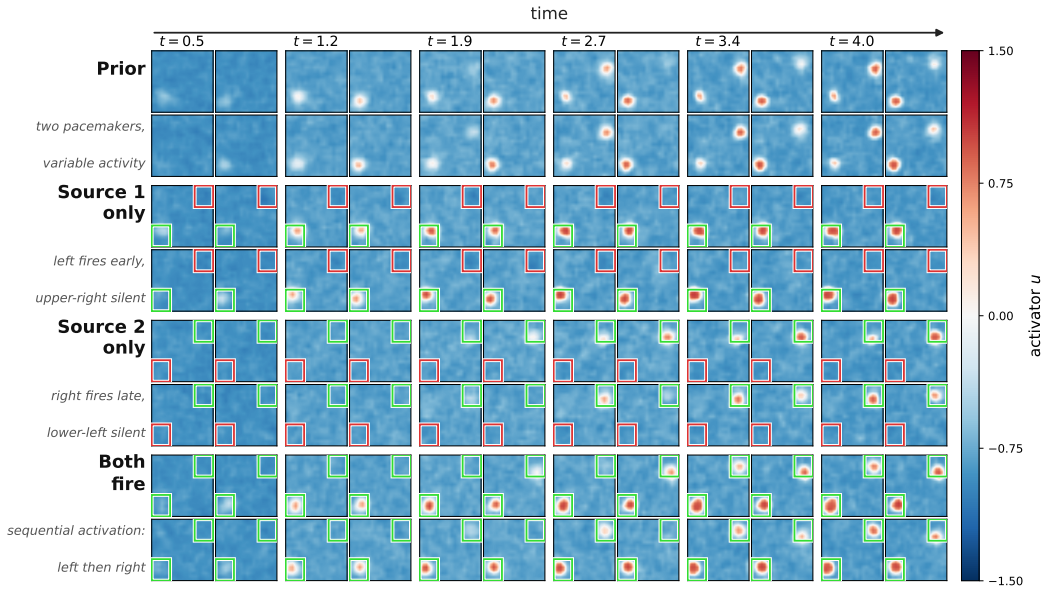


Figure 20: **Posterior samples from the FitzHugh–Nagumo excitable medium under three conditioning regimes, each showing four independent draws at six time snapshots**. *Prior:* unconditioned samples show variable activation of two spatially separated pacemakers (green boxes: active; red boxes: silent). *Source 1 only:* conditioning on the lower-left pacemaker firing early ($t \lesssim 1.3$) while the upper-right remains silent produces a single travelling wavefront originating from the lower-left corner. *Source 2 only:* the complementary condition suppresses the lower-left source and requires the upper-right to fire late ($t \gtrsim 1.8$), yielding a wavefront from the upper-right. *Both fire:* conditioning on sequential activation of both sources produces two distinct wavefronts propagating across the medium in succession.

boundary conditions. After each Euler–Maruyama step the field is saturated via $u \leftarrow 1.45 \tanh(u/1.45)$. The initial condition is

$$u_0(\mathbf{x}) = -0.15 + 0.65 \tanh\Big(\mathcal{S}^{(3)}(z_{\mathbf{x}}^{(0)})\Big),$$

where $\mathcal{S}^{(3)}$ denotes three passes of a box-filter smoother, which introduces spatial correlations and controls the characteristic length scale of the initial phase domains; $z_{\mathbf{x}}^{(0)} \sim \mathcal{N}(0, I)$ is a 4096-dimensional spatial latent with one scalar per grid point. The SDE noise is likewise reparameterised: at each of the $T = 30$ steps a spatially independent 64×64 Gaussian latent field is drawn and scaled by $\sigma\sqrt{\Delta t}$ to produce the Euler–Maruyama increment $\sigma\, \mathrm{d}W$. LatentFlow performs inference over the full latent vector $\mathbf{z}$ comprising both the initial-condition field and all SDE noise increments.

G.7.2 Conditions

We consider four conditioning regimes. Except for the sparse-sensor experiment, all conditions are imposed on the final time step $u(\cdot, t_{29})$ and implemented as equality constraints with observation noise σ_{obs}.

Sparse sensors. A 6×6 regular grid of sensor locations $\{(y_i, x_j)\}$ is placed uniformly across the domain (fractional positions $\{1/12, 3/12, 5/12, 7/12, 9/12, 11/12\}$ in each axis), giving 36 spatial sensors observed at each of 6 snapshot times $t \in \{t_0, t_6, t_{12}, t_{18}, t_{24}, t_{29}\}$, yielding $36 \times 6 = 216$ scalar observations. The condition is

$$u(\mathbf{x}_i, t_k) = u^*(\mathbf{x}_i, t_k) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_{\text{obs}}^2),\ \sigma_{\text{obs}} = 10^{-5},$$

where u^* is sampled from the prior as a ground truth.

Quadrant means. The spatial domain is divided into four quadrants $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$ (upper-left, upper-right, lower-left, lower-right). The condition prescribes the spatial mean of $u(\cdot, t_{29})$ in

each quadrant:

$$\frac{1}{|\mathcal{Q}_k|}\sum_{\mathbf{x}\in\mathcal{Q}_k} u(\mathbf{x}, t_{29}) = m_k^*, \quad (m_1^*, m_2^*, m_3^*, m_4^*) = (+0.10,\ -0.10,\ -0.10,\ +0.10),$$

with $\sigma_{\text{obs}} = 10^{-5}$, inducing a diagonal checkerboard mean pattern.

Annular ring. The domain is partitioned into three radially symmetric regions: a core disc of radius $r_{\text{in}} = 0.30$, an annular ring $r_{\text{in}} \le r \le r_{\text{out}} = 0.65$, and the exterior $r > r_{\text{out}}$. The condition prescribes the spatial mean in each region at t_{29}:

$$\bar{u}_{\text{ring}} = +0.15, \quad \bar{u}_{\text{core}} = -0.15, \quad \bar{u}_{\text{exterior}} = -0.15,$$

with $\sigma_{\text{obs}} = 10^{-4}$, producing a positive annulus surrounded by negative phases.

Phase front. The domain is divided into three vertical strips by boundaries at $x = \pm 0.30$. The condition prescribes the spatial mean of $u(\cdot, t_{29})$ in each strip:

$$\bar{u}_{\text{left}} = -0.20, \quad \bar{u}_{\text{centre}} = 0.0, \quad \bar{u}_{\text{right}} = +0.20,$$

with $\sigma_{\text{obs}} = 10^{-4}$, inducing a left-to-right phase gradient with a diffuse interface in the central strip.

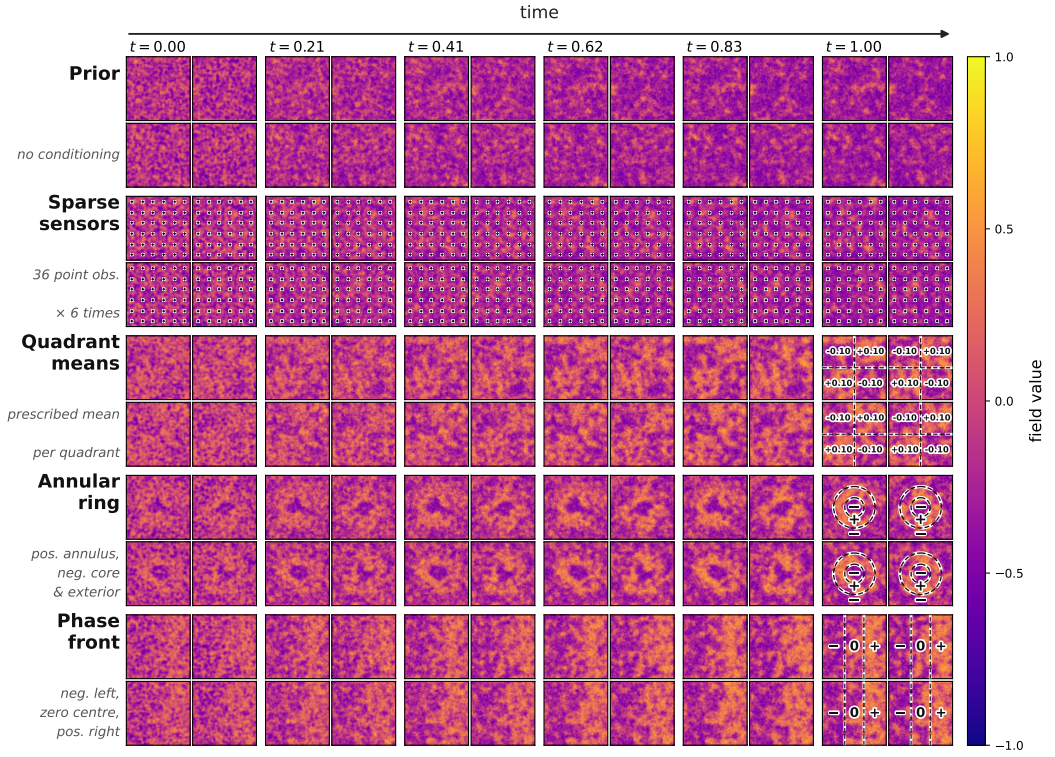


Figure 21: **Posterior samples from the Allen–Cahn phase-field model under four conditioning regimes, each showing four independent draws at six time snapshots.** Except for the sparse-sensor experiment, all conditions constrain statistics of the final field $u(\cdot, t_{29})$. *Prior:* unconditioned samples exhibit stochastic domain coarsening toward the ± 1 phases. *Sparse sensors:* conditioning on 36 point observations at each of 6 snapshot times (216 observations total) recovers the ground-truth field trajectory. *Quadrant means:* prescribing the spatial mean in each quadrant to a checkerboard pattern $(+0.10, -0.10, -0.10, +0.10)$ steers the final phase arrangement into the corresponding diagonal structure. *Annular ring:* prescribing a positive mean in an annular band $(0.30 \le r \le 0.65)$ and negative means in the core and exterior produces a ring-shaped positive domain surrounded by negative phase. *Phase front:* prescribing a left-to-right mean gradient $(-0.20, 0.0, +0.20)$ across three vertical strips ($|x| \le 0.30$ boundaries) induces a diffuse phase interface running vertically through the domain.

H Bayesian Parameter Inference: Experimental Details and Baselines

This appendix details the two parameter-inference experiments of Section 5: the Cauchy-convolution field and the stochastic Lotka–Volterra system. It also contains details of the latent-space MCMC baselines they are compared against.

H.1 Cauchy-convolution field

Prior and target. A Cauchy-convolution field on a 32×32 grid with a Matérn-3/2 convolution kernel. We use a unit output scale, which the map normalises out and is therefore not inferred. The lengthscale ℓ is the inferred parameter, with a log-normal prior of log-mean $\log 0.7$ and log-std 0.9; the ground truth is $\ell^\star = 0.1$. The latent is $\big(\xi_{\text{field}} \in \mathbb{R}^{1024}, \zeta_\ell \in \mathbb{R}\big)$. Since ℓ differs per sample, the convolution kernel is rebuilt for each latent from an analytic, differentiable batched Matérn gram.

Condition. The field is observed only through the *maximum* value within each of $K = 32$ irregular (Voronoi) regions.

Sampling. 40 posterior samples with the LatentFlow sampler, a single guidance draw per step, 2000 SDE steps, and a guidance-norm clip of 10^2.

H.2 Stochastic Lotka–Volterra

Prior and target. A geometric (log-space Euler–Maruyama) stochastic Lotka–Volterra predator–prey process on $N = 50$ time points over $[0, 5]$, diffusion scale 0.08. The four rates $(\alpha, \beta, \gamma, \delta)$ and the two initial populations are inferred. Each of these has a log-normal prior: the rates with log-means $\log(1.2, 1.6, 2.0, 0.2)$, which are deliberately misspecified against the truth $(2.2, 0.9, 1.5, 0.7)$, and log-std 0.35; and the initial populations with log-means $\log(2, 1)$ and log-std 0.3. The latent is $\big(\xi_{\text{traj}} \in \mathbb{R}^{2\times 50}, \zeta \in \mathbb{R}^6\big)$, with the six parameter coordinates shared across the two (prey, predator) channels.

Condition. Both populations are observed densely (every second point) within the first 30% of the window. That is, we consider an early-outbreak regime. The observation likelihood has scale $\sigma = 0.15$, which balances likelihood and prior. In particular, $\sigma \lesssim 0.1$ results in the likelihood dominating the prior, and subsequently the posterior for the rates to collapse.

Sampling. 40 posterior samples using the LatentFlow sampler, 2000 SDE steps, guidance-norm clip 2×10^2, and 100 Monte-Carlo guidance draws per step.

H.3 Latent-space MCMC baselines

Main baselines. We consider two MCMC samplers purpose-built for Gaussian priors. These methods are constructed to be dimension-robust; that is, their mixing does not degrade as the latent dimension grows. These methods are as follows.

- **Preconditioned Crank–Nicolson (pCN)** (Cotter et al., 2013). This sampler uses proposal $\xi' = \sqrt{1-\varrho^2}\,\xi + \varrho\,\zeta$, $\zeta \sim \mathcal{N}(0, I)$, which is reversible with respect to the $\mathcal{N}(0, I)$ prior. The acceptance ratio is $\min\big(1,\, L_C(\tilde{T}(\xi'))/L_C(\tilde{T}(\xi))\big)$.
- **Elliptical slice sampling (ES)** (Murray et al., 2010). This sampler draws an auxiliary $\nu \sim \mathcal{N}(0, I)$, forms the ellipse $\xi\cos\theta + \nu\sin\theta$, and shrinks an angular bracket until a likelihood threshold is met.

Each sampler is run as many parallel chains (one per batch element); for pCN the step size ϱ is adapted during burn-in toward a 0.25 acceptance rate by a Robbins–Monro update on $\log\varrho$ and then frozen. The specific chain settings are as follows:

Experiment	Sampler	Chains	Steps (burn-in)	Thin
Cauchy	pCN	64	6000 (3000)	30
	ES	32	400 (150)	5
Lotka–Volterra	pCN	128	8000 (2000)	20
	ES	128	2000 (500)	5

Additional NUTS baseline for Lotka–Volterra. As an independent, gold-standard verification we also sample the Lotka–Volterra rate posterior with the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014), as implemented in NumPyro (Phan et al., 2019). In this case we use the native representation of the process, rather than the latent representation . This is possible since, in log-population coordinates, the geometric Euler–Maruyama dynamics are conditionally Gaussian. Thus the joint density of the rates and the log-population trajectory is a product of Gaussian transition and observation factors, with no recursive simulation to differentiate through. In this case we run 4 chains of 1000 warm-up and 1500 sampling iterations each, with a dense mass matrix over the four rate coordinates, target acceptance 0.9, and a maximum tree depth of 10, initialised from the observations. The sampler converges cleanly in roughly 6 s on CPU, and its rate marginals agree with LatentFlow and the latent-space (pCN, elliptical slice) posteriors, including the weakly-identified γ. It is worth emphasising that our ability to run NUTS here is specific to Lotka–Volterra (and the conditioning statement). In the Cauchy example, there is no such tractable joint density, so no analogous parameter-space sampler is available.

Diagnostics. We track, per parameter, the posterior median with 90% credible interval and track the split-$\hat{R}$ statistic across chains. In particular, each chain is split in half, so within-chain non-stationarity registers as between-half disagreement. A reference is treated as converged when $\hat{R} \leq 1.2$.

Reproducibility. Repeating the guided inference under five sampler seeds on the same fixed problem yields highly stable posterior medians (standard deviation below 0.1 for every Lotka–Volterra rate and below 0.005 for the Cauchy lengthscale).

Results. The results are shown in Tab. 17. For the Lotka–Volterra example, there is close agreement between all of the methods, providing further evidence that LatentFlow generates samples from the true posterior; in this case the joint posterior over the process and the parameters. Meanwhile, for the Cauchy example, all of the other methods fail to converge. In particular, pCN achieves $\hat{R} = 12$, and ESS achieves $\hat{R} = 7$, even at a matched or greater computational cost. Meanwhile, LatentFlow generates samples from a posterior which, as expected, concentrates around the true (i.e., data generating) parameter.

		Truth	LatentFlow	pCN	ES	NUTS
Cauchy	ℓ	0.10	0.11 [0.07, 0.22]	×	×	×
	Time ℓ		79s	128s	96s	×
Lotka–Volterra	α	2.20	2.19 [1.59, 2.71]	2.17 [1.61, 2.88]	2.18 [1.63, 2.85]	2.16 [1.62, 2.85]
	β	0.90	0.89 [0.63, 1.12]	0.93 [0.65, 1.29]	0.93 [0.65, 1.27]	0.92 [0.64, 1.27]
	γ	1.50	1.13 [0.75, 1.74]	1.18 [0.73, 1.86]	1.20 [0.75, 1.84]	1.18 [0.75, 1.79]
	δ	0.70	0.55 [0.46, 0.73]	0.57 [0.43, 0.75]	0.58 [0.44, 0.75]	0.57 [0.44, 0.74]
	Time		23s	24s	101s	6s

Table 17: LatentFlow against three MCMC references — pCN and ESS in our innovation space and, when applicable, NUTS in the parameter space. Each entry is the posterior median with 90% credible interval. On Lotka–Volterra, all three references converge and match the guided posterior; on the Cauchy region-maximum condition pCN and ESS fail to mix, NUTS is not applicable, and only LatentFlow provides posterior samples.