Conditioning Gaussian Processes on Almost Anything

Henry B. Moss *
Lancaster University

Lachlan Astfalck*
University of New South Wales

Thomas Cowperthwaite
University of Cambridge

Colin Doumont
University of Tübingen
Tübingen AI Center

Sam Willis
University of Cambridge

Philipp Hennig
University of Tübingen
Tübingen AI Center

Christopher Nemeth
Lancaster University

Andrew Zammit-Mangion
University of New South Wales

Abstract

Gaussian processes (GPs) offer a principled probabilistic model over functions, but exact inference is restricted to the linear-Gaussian regime. We establish an explicit equivalence between GPs and a class of linear diffusion models, recasting predictive sampling as an ODE with closed-form Gaussian dynamics and a likelihood-dependent guidance term that admits a simple Monte Carlo approximation. In the linear-Gaussian setting, we recover standard GP conditioning exactly; beyond conjugacy, the same machinery handles any conditioning statement admitting point-wise likelihood evaluation — including non-linear physics, and, for the first time, natural language via large language models. Whitening isolates the irreducible non-Gaussian dynamics, minimising Wasserstein-2 transport cost and eliminating numerical stiffness. The result is a general-purpose GP inference scheme requiring no bespoke derivations. Together, these results provide a general mechanism for incorporating the full richness of real-world knowledge as conditioning information, opening a new frontier for the probabilistic modelling of real-world problems.

1 Introduction

Gaussian processes (GPs) provide a fully specified probabilistic model over functions with closed-form conditioning under linear Gaussian observations, offering a rare combination of flexibility and mathematical exactness [1]. Beyond the linear-Gaussian regime, exact conditioning is lost: non-Gaussian likelihoods and non-linear constraints force reliance on approximate inference schemes such as Laplace approximations [2], expectation propagation [3], or variational methods [4], which are often computationally demanding, sensitive to parametrisation, and difficult to generalise across data modalities. This motivates a framework for conditional GP sampling that treats the problem in full generality, extending beyond conjugacy without bespoke approximations.

Our perspective draws on recent advances in diffusion modelling [5, 6] and flow-matching [7, 8, 9], which transport samples from a simple reference distribution (typically Gaussian) through a sequence of intermediate distributions to match a target law [10]. In diffusion modelling, this transport is described via the reverse-time SDE of a noising process [6]; in flow-matching, a learnt velocity field induces an ODE pushing the reference to the target [7]. In both cases, sampling is realised as an evolution along a path in distribution space. This has led to the realisation that conditioning

*Joint first author. Correspondence to henry.moss@lancaster.ac.uk and l.astfalck@unsw.edu.au

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

can be achieved by modifying the prior sampling dynamics at test time so that trajectories are biased towards regions satisfying the conditioning statement [11]. Exactness is contingent on three approximations: (i) accuracy of the learned score, (ii) quality of the conditional score along the path, and (iii) discretisation error from the SDE/ODE solver. Function space extensions exist, for example FunDiff [12] and DDO [13]; however, these still train a neural approximation of the score and remain subject to the same approximation errors.

The main contribution of this work is the realisation that GPs are an exact closed-form instance of diffusion and flow-matching models (see Section 3). General GP conditioning on arbitrary conditions may then be written as the modification of an underlying analytical transport, and existing conditioning techniques from the diffusion literature can be incorporated (Section 4). Of the three sources of error mentioned above, the first is eliminated entirely: the GP prior is available in closed form and no score network need be learnt. We prove that the remaining two sources, Monte Carlo guidance estimation and solver discretisation, are independently controllable, yielding predictive samples at prescribed fidelity with a fixed per-sample cost and without the mixing-time dependencies of MCMC or the parametrisation sensitivities of variational inference.

Our contributions offer a new frontier for the GP modelling of practical real-world problems:

- **GP–diffusion equivalence.** We establish an explicit equivalence between GPs and a class of linear diffusion models, providing an alternative transport view of GP sampling.
- **Conditioning on anything.** We show that standard linear GP conditioning arises as a special case of test-time guidance, and extend this to arbitrary non-linear, non-Gaussian conditioning via a Monte Carlo approximation requiring only point-wise likelihood evaluations.
- **Whitening is the optimal parameterisation.** We show whitening minimises Wasserstein-2 transport cost, with numerical stability governed by the covariance condition number.
- **A unified GP sampler.** We apply our sampler (FLOWGP) across applications including constrained regression, physics-informed modelling, Bayesian optimisation, and conditioning GPs on natural language via large language models (see Figure 1).
- **Computational efficiency.** FLOWGP runtimes range from milliseconds for monotonic bounded regression, to less than 5 seconds for LLM- and physics-constrained generation. Code is available at `github.com/FlowInference/flow-inference`.

2 A primer on Gaussian processes

Let $f \in \mathcal{H}$ be a random function in some function space $\mathcal{H}$ indexed by locations $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$. The defining property of a GP is that for any finite collection of m test inputs $\mathbf{X}_* \in \mathcal{X}^m$, the random vector $\boldsymbol{f}_0 \in \mathbb{R}^m$ of evaluations of $f(\cdot)$ at $\mathbf{X}_*$ follows a multivariate Gaussian distribution $\boldsymbol{f}_0 \mid \mathbf{X}_* \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$, where $\mathbf{m}_* = \mu(\mathbf{X}_*) \in \mathbb{R}^m$ and covariance matrix $\mathbf{K}_{**} := k(\mathbf{X}_*, \mathbf{X}_*) \in \mathbb{R}^{m \times m}$ are induced by a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We depart from the conventional notation of $\boldsymbol{f}_*$, instead using $\boldsymbol{f}_0$ to reflect both (i) the standard statistical interpretation of a "true" parameter and (ii) the notation of generative modelling, in which $t = 0$ typically marks the data-generating endpoint.

One of the most desirable features of GPs is that they remain GPs under Gaussian conditioning with a bounded linear operator [14]. Let $\mathcal{L} : \mathcal{H} \to \mathbb{R}^n$ be a linear operator acting on the sample paths of $f(\cdot)$ to generate data at locations $\mathbf{X}_n \in \mathcal{X}^n$. Its action on $f(\cdot)$ can be represented as a matrix $\mathbf{L} \in \mathbb{R}^{n \times m}$ acting on a discretisation of $f(\cdot)$, $\boldsymbol{f}_0$, which also constitutes our inferential target. Now consider the following data model for a noisy realisation $\boldsymbol{y}$,

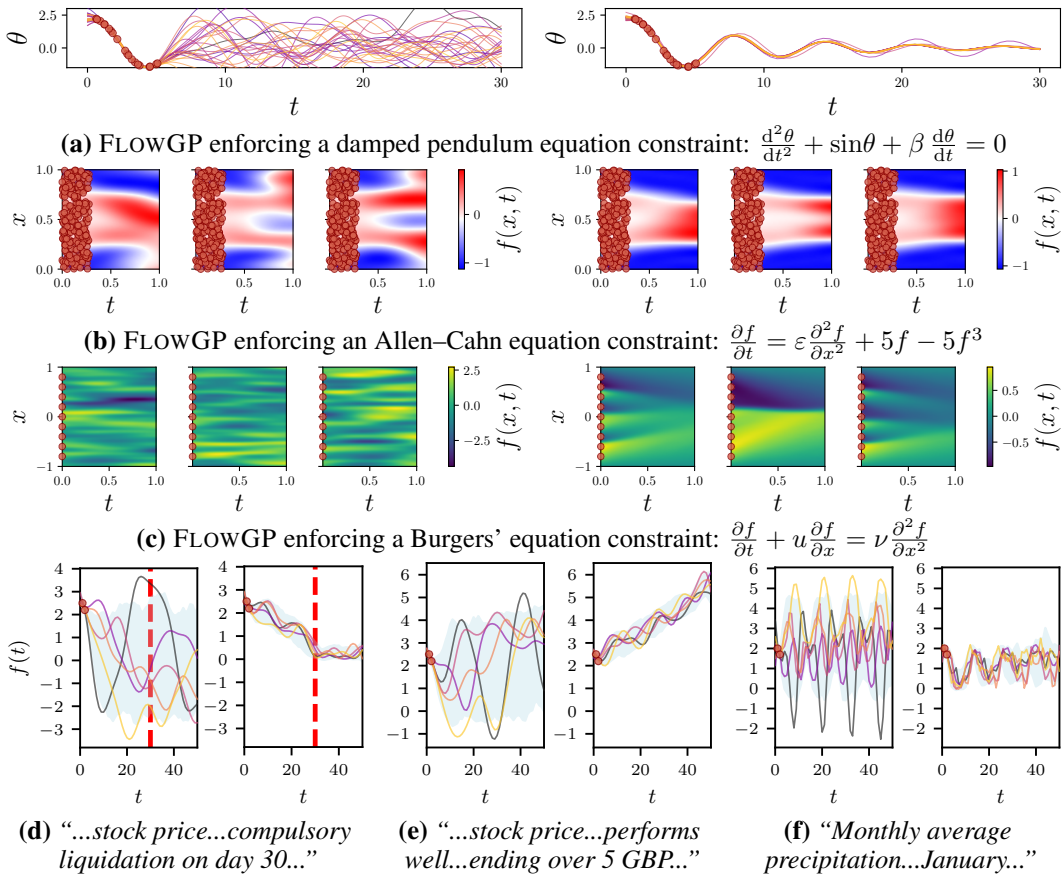
$$\boldsymbol{y} = \mathbf{L}\boldsymbol{f}_0 + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_n, \boldsymbol{\Gamma}), \tag{1}$$

where $\mathbf{0}_n$ denotes the n-vector of zeros, and where $\boldsymbol{\Gamma}$ denotes the covariance matrix of the noise $\boldsymbol{\varepsilon}$ (e.g. $\boldsymbol{\Gamma} = \sigma^2 \mathbf{I}_n$). The GP predictive distribution $\boldsymbol{f}_0 \mid \mathcal{D} \sim \mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$ has mean and covariance

$$\mathbf{m}_{*|\boldsymbol{y}} = \mathbf{m}_* + \mathbf{K}_{**}\mathbf{L}^\intercal \left(\mathbf{L}\mathbf{K}_{**}\mathbf{L}^\intercal + \boldsymbol{\Gamma}\right)^{-1} (\boldsymbol{y} - \mathbf{L}\mathbf{m}_*), \tag{2}$$

$$\mathbf{K}_{**|\boldsymbol{y}} = \mathbf{K}_{**} - \mathbf{K}_{**}\mathbf{L}^\intercal \left(\mathbf{L}\mathbf{K}_{**}\mathbf{L}^\intercal + \boldsymbol{\Gamma}\right)^{-1} \mathbf{L}\mathbf{K}_{**}^\intercal, \tag{3}$$

where $\mathcal{D} = \{\mathbf{X}_n, \boldsymbol{y}\}$. Key applications of Gaussian processes conditioned on linear operators in the Gaussian setting include linear inverse problems [15]; making inference from derivative observations [16, 17]; data assimilation [18, 19]; and problems involving linear constraints [20].



(a) FLOWGP enforcing a damped pendulum equation constraint: $\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \sin\theta + \beta\,\frac{\mathrm{d}\theta}{\mathrm{d}t} = 0$

(b) FLOWGP enforcing an Allen–Cahn equation constraint: $\frac{\partial f}{\partial t} = \varepsilon\frac{\partial^2 f}{\partial x^2} + 5f - 5f^3$

(c) FLOWGP enforcing a Burgers' equation constraint: $\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = \nu\frac{\partial^2 f}{\partial x^2}$

(d) *"...stock price...compulsory liquidation on day 30..."*

(e) *"...stock price...performs well...ending over 5 GBP..."*

(f) *"Monthly average precipitation...January..."*

Figure 1: **(left of each pair)** Samples from a GP conditioned on observations (red dots) and **(right of each pair)** samples from FLOWGP including additional information about non-linear physics via known differential equations **(a-c)** and natural language descriptions via an LLM-based likelihood **(d-f)**. In each case, the unconstrained GP produces statistically coherent but semantically uninformed samples, whilst FLOWGP produces samples that are also faithful to known physics or semantic content of the text prompt. See Section 6 for full experimental details.

Techniques for sampling from GPs. The standard approach is to transform white noise $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ via the mean-scale transformation: $\boldsymbol{f}_0 = \mathbf{m}_* + \mathbf{K}_{**}^{1/2}\boldsymbol{z}$, where $\mathbf{K}_{**}^{1/2}$ denotes the matrix square root, raising a substantial $\mathcal{O}(m^3)$ factorisation cost. The SPDE approach [21, 22] and its Hilbert space extensions [23] specify a Gaussian Markov random field with sparse precision matrix, whose Cholesky factorization costs roughly $\mathcal{O}(m^{3/2})$; pathwise conditioning [24] constructs differentiable predictive sample paths as a prior sample plus a deterministic update; sparse approximations [25, 4, 26, 27] and computation-aware iterative solvers [28, 29] further reduce the cost of drawing samples from large-scale GP predictive distributions. These methods and approximations are compatible with FLOWGP (Section 4) and provide avenues for scalability.

Physics-informed Gaussian processes. Kernel-based approaches encoding physics through the covariance structure are restricted to linear PDEs [30, 31, 32, 33, 34, 35]. Modern approximate methods extend to non-linear PDEs — PHYSS [36] and AUTOIP [37] via variational inference, and via kernel smoothing [38], though the latter yields only point estimates. All of these methods require bespoke derivations tailored to each PDE. Probabilistic ODE solvers [39, 40] model solutions as GPs with calibrated discretisation uncertainty and generalise linear PDE solvers [14], but extensions to non-linear PDEs remain limited.

3 Sampling GPs under linear-Gaussian conditioning by solving an ODE

We present a new perspective on GP sampling and derive an ODE whose dynamics produce the samples from the target Gaussian distribution. Consider the set of vectors $\{\boldsymbol{f}_t\}_{t\in[0,1]}$, each defined by mapping white noise samples $\boldsymbol{z}$ via the flow map

$$\boldsymbol{f}_t = \Phi(t, \boldsymbol{z}) := \mathbf{b}(t) + \mathbf{A}(t)^{1/2}\boldsymbol{z}, \quad t \in [0, 1], \tag{4}$$

where $\mathbf{A}(t) := \alpha^2(t)\mathbf{K}_{**} + (1 - \alpha^2(t))\mathbf{I}_m$ and $\mathbf{b}(t) := \alpha(t)\mathbf{m}_*$. We require that $\alpha(t)$ is a monotonically decreasing and differentiable function, with $\alpha(0) = 1$, and $\alpha(1) \to 0^+$ to ensure that $\boldsymbol{f}_0 \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$ and $\boldsymbol{f}_1 \to \boldsymbol{z}$. Differentiating the flow map $\Phi(\cdot, \cdot)$ in (4) with respect to t, we obtain the following linear time-varying ODE,

$$\boxed{\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t) \coloneqq -\frac{1}{2}\beta(t)\mathbf{A}(t)^{-1}\mathbf{b}(t) - \frac{1}{2}\beta(t)\left(\mathbf{I}_m - \mathbf{A}(t)^{-1}\right)\boldsymbol{f}_t, \quad t \in [0, 1],} \tag{5}$$

where $v(\boldsymbol{f}_t, t)$ is the velocity field, $\boldsymbol{f}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ and $\beta(t) = -2\frac{\alpha'(t)}{\alpha(t)} = -2\frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}$. Across all the results in this paper, we use a simple linear schedule for $\beta(t)$ (see Appendix F.1). The ODE (5) yields a strategy for sampling from $\boldsymbol{f}_0$: starting from white noise $\boldsymbol{f}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ and numerically integrating the ODE (5) backwards in time from $t = 1$ to $t = 0$ produces samples from the target distribution at $t = 0$ (see Figure 2).

3.1 An alternative perspective: sampling from Gaussian processes with a diffusion

The time-t marginals $p(\boldsymbol{f}_t), t \in [0, 1]$ induced by the ODE in (5) are the same as those of a variance-preserving (VP) SDE [6] of the form

$$\mathrm{d}\boldsymbol{f}_t = -\frac{1}{2}\beta(t)\,\boldsymbol{f}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{W}_t, \quad t \in [0, 1], \tag{6}$$

where $\boldsymbol{W}_t$ is an m-dimensional Wiener process. To see this, first observe that the conditional distributions $p(\boldsymbol{f}_t \mid \boldsymbol{f}_0)$ generated by the SDE in (6) are, for all $t \in [0, 1]$, the same as those generated by the perturbation model [7]

$$\boldsymbol{f}_t = \alpha(t)\boldsymbol{f}_0 + \sqrt{1 - \alpha^2(t)}\boldsymbol{z}, \quad t \in [0, 1], \tag{7}$$

where $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$. This can be verified by showing that the conditional expectations and variances of the solution to (6) match those of (7) for each t. It follows that for a distribution $p(\boldsymbol{f}_0)$, the SDE in (6) induces the same marginals

$$p(\boldsymbol{f}_t) = \int p(\boldsymbol{f}_t \mid \boldsymbol{f}_0)p(\boldsymbol{f}_0)\,\mathrm{d}\boldsymbol{f}_0, \quad t \in [0, 1], \tag{8}$$

as (7). Second, it is immediate that (5) generates the same marginals $\boldsymbol{f}_t \sim \mathcal{N}(\mathbf{b}(t), \mathbf{A}(t)), t \in [0, 1]$, as (7) when $\boldsymbol{f}_0 \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$, and therefore the same as those from the SDE in (6). We may therefore replace simulation of the SDE with numerical integration of the ODE in (5) when sampling from these marginals.

In most applications of diffusion sampling, (8) is approximated with a neural network as a function of t, however these marginals are available in closed-form when the distribution of $\boldsymbol{f}_0$ is induced from a GP. Moreover, the conditional distribution $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t)$ of the SDE is also available; in Appendix A we show that for $\boldsymbol{f}_0 \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$,

$$\boldsymbol{f}_0 \mid \boldsymbol{f}_t \sim \mathcal{N}\left(\mathbf{m}_* + \alpha(t)\mathbf{K}_{**}\mathbf{A}(t)^{-1}\left(\boldsymbol{f}_t - \alpha(t)\mathbf{m}_*\right), \mathbf{K}_{**} - \alpha^2(t)\mathbf{K}_{**}\mathbf{A}(t)^{-1}\mathbf{K}_{**}\right). \tag{9}$$

This property will be particularly useful when computing with non-linear guidance in Section 4.

The above discussion showed that the probabilistic flow (5) generates margins that are the same as that of the SDE (6). We can also derive the probability flow (5) directly from the SDE. It is well known [41] that the probability-flow ODE associated with the SDE (6) is specified by the score function $s(\boldsymbol{f}_t, t) = \nabla_{\boldsymbol{f}_t}\log p(\boldsymbol{f}_t)$ via

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = -\frac{1}{2}\beta(t)\left(\boldsymbol{f}_t + s(\boldsymbol{f}_t, t)\right); \tag{10}$$

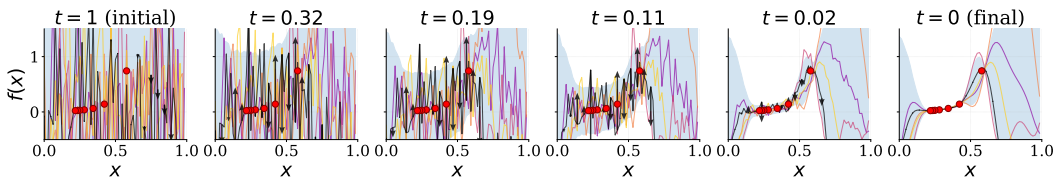


Figure 2: Generating samples from the GP predictive distribution when conditioning on Gaussian observations (red dots). Sample trajectories (lines), velocity field (arrows), and empirical quantiles of $p(\boldsymbol{f}_t)$ (shading) are shown at different times. The flow interpolates white noise (left, $t = 1$) to samples from the predictive distribution (right, $t = 0$).

see [10] for a derivation. In our Gaussian setting, the score is simply $s(\boldsymbol{f}_t, t) = -\mathbf{A}(t)^{-1}(\boldsymbol{f}_t - \mathbf{b}(t))$; substituting this into (10) recovers the linear ODE in (5). We use (10) directly in Sections 3.2 and 4.1 to implement linear-Gaussian and nonlinear-non-Gaussian guidance, respectively. The ODE (10) also admits a natural flow matching interpretation: the interpolant (7) is precisely the affine Gaussian interpolant of stochastic interpolant flow matching [8], making FLOWGP an exact closed-form instance of flow matching for Gaussian targets.

3.2 Sampling under linear Gaussian conditioning as score-based guidance

Recall the linear Gaussian data model in (1) that yields the predictive distribution $\boldsymbol{f}_0 \mid \mathcal{D} \sim \mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$. As $\boldsymbol{f}_0 \mid \mathcal{D}$ remains Gaussian, we may substitute $\mathbf{b}_{|\boldsymbol{y}}(t) := \alpha(t)\mathbf{m}_{*|\boldsymbol{y}}$ and $\mathbf{A}_{|\boldsymbol{y}}(t) := \alpha^2(t)\mathbf{K}_{**|\boldsymbol{y}} + (1 - \alpha^2(t))\mathbf{I}_m$ for $\mathbf{b}(t)$ and $\mathbf{A}(t)$ in (5) to yield the ODE

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D}) := -\frac{1}{2}\beta(t)\mathbf{A}_{|\boldsymbol{y}}(t)^{-1}\mathbf{b}_{|\boldsymbol{y}}(t) - \frac{1}{2}\beta(t)\left(\mathbf{I}_m - \mathbf{A}_{|\boldsymbol{y}}(t)^{-1}\right)\boldsymbol{f}_t, \quad t \in [0, 1], \tag{11}$$

which can be used to generate samples from the predictive distribution $\boldsymbol{f}_0 \mid \mathcal{D} \sim \mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$ (see Figure 2), with marginals $\boldsymbol{f}_t \mid \mathcal{D} \sim \mathcal{N}(\mathbf{b}_{|\boldsymbol{y}}(t), \mathbf{A}_{|\boldsymbol{y}}(t))$.

Alternatively, one can start off from the probability flow ODE representation in (10) given by

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D}) := -\frac{1}{2}\beta(t)\left(\boldsymbol{f}_t + \nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D})\right). \tag{12}$$

The conditional score admits the decomposition

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}) = \underbrace{\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t)}_{\text{unconditional GP}} + \underbrace{\nabla_{\boldsymbol{f}_t} \log p(\mathcal{D} \mid \boldsymbol{f}_t)}_{\text{linear conditioning/guidance}}, \tag{13}$$

where the first term corresponds to the GP model (Section 3) and the second term encodes the conditioning information. While this recovers standard GP conditioning in a different guise, the decomposition reveals that conditioning can be implemented by augmenting the drift of the probability flow ODE with a *guidance term* $\nabla_{\boldsymbol{f}_t} \log p(\mathcal{D} \mid \boldsymbol{f}_t)$ [42]. Importantly, this framework provides a path forward to relax assumptions of linearity and Gaussianity in our conditioning statements.

4 FLOWGP: sampling GPs under non-linear and non-Gaussian conditioning

We now demonstrate the key utility of our ODE viewpoint of GP sampling: it enables efficient sampling from the GP predictive distribution arising from arbitrary non-linear non-Gaussian likelihoods. Specifically, we target

$$p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \propto \underbrace{p(\boldsymbol{f}_0)\, p(\mathcal{D} \mid \boldsymbol{f}_0)}_{\propto\, p(\boldsymbol{f}_0 \mid \mathcal{D})}\, p(\mathcal{C} \mid \boldsymbol{f}_0, \mathcal{D}), \tag{14}$$

where $\mathcal{D}$ denotes the information that admit a linear closed-form Gaussian update, and $\mathcal{C}$ denotes non-linear and non-Gaussian contributions, assumed conditionally independent given $\boldsymbol{f}_0$. We assume the GP prior and $\mathcal{C}$ are broadly compatible; if the prior assigns negligible probability to the region satisfying $\mathcal{C}$, sample quality may degrade and so kernel hyperparameters should be reasonably well-specified before applying FLOWGP; see Section 7.

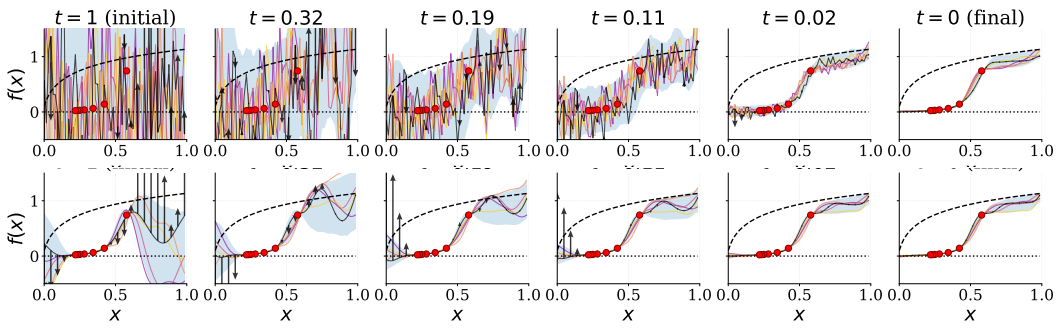


Figure 3: Extension of Figure 2 to non-Gaussian and non-linear likelihoods. **Top row**: Generating samples from the predictive distribution when conditioning on Gaussian observations (red dots) together with non-linear constraints enforcing boundedness and monotonicity (dashed lines). **Bottom row**: Generating samples from the same predictive distribution using the whitened formulation, in which the Gaussian dynamics are removed, producing smoother trajectories and avoiding the stiff behaviour present in the original formulation.

4.1 Sampling from the predictive distribution via an ODE

Section 3.2 outlines two admissible ODE-based methods for sampling from the predictive distribution under linear Gaussian conditioning: differentiating closed-form flow maps and using the probability flow (12). In general, no analytically-tractable flow map exists when conditioning includes $\mathcal{C}$, but the representation (10) offers a way forward to generate the required samples.

Consider a general conditioning set $\mathcal{C}$ and the sequence of marginal distributions obtained by incrementally noising our target conditional distribution

$$p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \int p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \, \mathrm{d}\boldsymbol{f}_0 \quad \text{where} \quad p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) = \mathcal{N}(\alpha(t)\boldsymbol{f}_0, (1-\alpha^2(t))\mathbf{I}_m).$$

By Bayes' rule, the score function of $p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C})$ decomposes as

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \underbrace{\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D})}_{\text{linear update in (13)}} + \underbrace{\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D})}_{\text{non-linear guidance}}. \tag{15}$$

This score function is the sum of the score of the linear update and an additional term encoding the non-linear and non-Gaussian contributions. Incorporating this term into the ODE (10) yields

$$\boxed{\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D}) - \frac{1}{2}\beta(t)\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D})} \tag{16}$$

where $\boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D})$ is the vector field (11) producing samples from the linear-Gaussian part of the predictive distribution, $p(\boldsymbol{f}_0 \mid \mathcal{D})$. If $\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D})$ is available, the flow can therefore be adjusted to generate samples from the full predictive distribution that also involves the nonlinear non-Gaussian components $\mathcal{C}$. Therefore, the ODE (16) yields a strategy for sampling from the predictive distributions of GPs that are not available in closed form: starting from white noise $\boldsymbol{f}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ and numerically integrating ODE (16) from $t = 1$ to $t = 0$ produces samples $\boldsymbol{f}_0$ from the target predictive distribution (see Figure 3).

4.2 Calculating the guidance term by Monte Carlo

Unlike $\boldsymbol{v}(\boldsymbol{f}_t, t \mid \mathcal{D})$, the nonlinear guidance term $\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D})$ appearing in (16) is generally intractable. However, since

$$\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) = \nabla_{\boldsymbol{f}_t} \log \int p(\mathcal{C} \mid \boldsymbol{f}_0) \, p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}) \, \mathrm{d}\boldsymbol{f}_0, \tag{17}$$

and $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ is a known Gaussian distribution, we can estimate it using a Monte Carlo approximation that only requires point-wise evaluation of the likelihood $p(\mathcal{C} \mid \boldsymbol{f}_0)$ and its score $\nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0)$.

For a sample of size S,

$$\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) \approx \alpha(t)\mathbf{K}_{**|\boldsymbol{y}}\mathbf{A}_{|\boldsymbol{y}}^{-1}(t) \sum_{i=1}^{S} \bar{w}^{(i)} s^{(i)}, \quad \text{for} \quad \bar{w}^{(i)} = w^{(i)} / \sum_{r=1}^{S} w^{(r)} \tag{18}$$

where $\boldsymbol{f}_0^{(i)} \sim p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$, and $w^{(i)} \propto p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$, $s^{(i)} = \nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$ for $i = 1, \ldots, S$. See Appendix B for derivation and implementation details. If $\nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0)$ is unavailable, a gradient-free estimate may be obtained via Fisher's identity (Appendix C).

Convergence and weight collapse. The estimator (18) is a self-normalised importance sampler. It is consistent, with a bias of order $\mathcal{O}(S^{-1})$, which vanishes faster than the $\mathcal{O}(S^{-1/2})$ standard Monte Carlo error under standard moment conditions on the importance weights. In high dimensions, self-normalised estimators can in principle suffer from *weight collapse* onto a single sample. Note that our construction mitigates this risk as the covariance of $\boldsymbol{f}_0|\boldsymbol{f}_t, \mathcal{D}$ remains a substantial fraction of $\mathbf{K}_{**|\boldsymbol{y}}$ for most of the integration interval, providing a broad proposal. Equivalently, the guidance can be read as a smoothed likelihood: at time t the flow follows a modulated likelihood smoothed by the conditional $p(\boldsymbol{f}_0|\boldsymbol{f}_t, \mathcal{D})$ resulting in heavy smoothing at large values of t and sharpening to the exact likelihood at terminal time $t = 0$. We acknowledge that collapse may nonetheless arise in high-dimensional settings and under tight conditions; developing adaptive remedies is an important direction for future work.

Alternative guidance approximations. In standard diffusion guidance, $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ is rarely available in closed form, motivating point-estimate methods such as DPS [43] and MPGD [44]. More sophisticated schemes incorporate higher-order information at additional computational cost [45]. Remarkably, in Appendix K we show that our simple Monte Carlo estimator outperforms DPS and MPGD even at $S = 1$, raising broader questions about the suitability of point-estimate approximations of guidance in diffusion sampling.

5 Practical and theoretical considerations

5.1 A natural parameterisation through whitening

Discretising the ODE in (16) over T steps is challenging when the covariance is poorly conditioned as the dynamics become stiff, which forces small step sizes. In Appendix D, we show that stiffness is governed by the condition number $\kappa(\mathbf{K}_{**|\boldsymbol{y}})$, which also controls numerical stability in standard GP sampling. In particular, Theorem D.1 establishes that under rescaling and in the regime $\lambda_{\min}(\mathbf{K}_{**|\boldsymbol{y}}) \to 0$, stiffness has two-sided asymptotic equivalence to $\kappa(\mathbf{K}_{**|\boldsymbol{y}})$, i.e., stiffness $\asymp \kappa(\mathbf{K}_{**|\boldsymbol{y}})$. This regime arises, for instance, under increasing data in a fixed domain. Increased stiffness leads to numerical instabilities that we seek to avoid.

This challenge is resolved by whitening. Define $\hat{\boldsymbol{f}}_0 \coloneqq \mathcal{W}\boldsymbol{f}_0 = \mathbf{K}_{**|\boldsymbol{y}}^{-1/2}(\boldsymbol{f}_0 - \mathbf{m}_{*|\boldsymbol{y}})$, under which $\hat{\mathbf{A}}(t) = \mathbf{I}_m$ and $\hat{\boldsymbol{b}}(t) = \mathbf{0}$, so the linear Gaussian velocity term vanishes, leaving only the non-linear guidance contribution

$$\boxed{\frac{\mathrm{d}\hat{\boldsymbol{f}}_t}{\mathrm{d}t} = \cancelto{0}{\boldsymbol{v}(\hat{\boldsymbol{f}}_t, t \mid \mathcal{D})} - \frac{1}{2}\beta(t)\nabla_{\hat{\boldsymbol{f}}_t} \log p(\mathcal{C} \mid \hat{\boldsymbol{f}}_t, \mathcal{D}),} \tag{19}$$

and the conditional distribution from the underlying whitened SDE becomes $\hat{\boldsymbol{f}}_0 \mid \hat{\boldsymbol{f}}_t, \mathcal{D} \sim \mathcal{N}(\alpha(t)\hat{\boldsymbol{f}}_t, (1 - \alpha^2(t))\mathbf{I}_m)$. Predictive samples are recovered from the solutions to (19) via $\boldsymbol{f}_0 = \mathcal{W}^{-1}\hat{\boldsymbol{f}}_0$.

Why whitening? The original dynamics of (16) couple two operations: transporting white noise to a correlated Gaussian, and steering samples towards $p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C})$ via non-linear guidance. Whitening handles the first explicitly, so the whitened ODE retains only the irreducible non-Gaussian dynamics. This decomposition is also optimal from a transport perspective: Theorem D.3 and Corollary D.4 show that whitening minimises the upper bound on the squared Wasserstein–2 distance $W_2^2(p(\boldsymbol{f}_0), p(\boldsymbol{f}_1))$ for all schedules $\beta(t)$.

5.2 Solving the resulting ODE

In practice FLOWGP incurs two sources of error: (i) the guidance approximation, and (ii) ODE solver discretisation. Theorem E.1 shows these decompose cleanly into a guidance error governed by log-likelihood curvature and a discretisation error governed by solver order and step size, which need not compound whenever the likelihood is sufficiently regular. For log-concave likelihoods, including classification, Poisson regression, and convex constraints, guidance error remains bounded or accumulates at most linearly, controllable by increasing S or reducing step size. Non-log-concave settings, such as mixture models or LLM-based likelihoods, admit only a worst-case exponential bound, though this is likely loose in practice: empirical results in Figure 1 are well-behaved despite this classification. Understanding precisely when this regularisation suffices is an important direction for future work. A full error analysis is given in Appendix E.

Non-differentiable likelihoods. A conclusion from Theorem E.1 is that, in order to practically solve FLOWGP's ODE under genuinely non-differentiable conditions such as binary or equality constraints, we require a smooth surrogate. Therefore, we propose binary conditions (e.g. inequality constraints) to be wrapped in a probit likelihood with small bandwidth ν, and equality conditions are replaced by a Gaussian likelihood with small variance σ, each recovering the original constraint in the limit.

6 Experimental details

We now provide details for a number of experimental settings, each chosen to demonstrate a key benefit of FLOWGP. First, we establish competitiveness with bespoke constrained regression methods (Section 6.1). Second, we show that our framework occupies a previously unoccupied position in physics-informed GP modelling, handling non-linear PDEs without bespoke kernels at a fraction of the cost (Section 6.2). We then demonstrate that encoding structural knowledge into the GP predictive distribution via FLOWGP improves a down-stream decision-making in Bayesian optimisation (Section 6.3). Next, we consider a controlled setting where the likelihood function is sharp. Finally, we explain how we used FLOWGP in tandem with large language models to condition on natural language — a capability beyond the reach of any existing GP method (Section 6.5).

Implementation details. We employ a linear diffusion schedule $\beta(t) = 10^{-5} + 10t$ and solve the resulting ODE via a simple Euler scheme for 1000 time steps. The SNR-uniform time discretisation of [46] is used throughout (see Appendix F.2); this concentrates steps near $t = 0$ where Theorem D.1 identifies stiffness to be maximal, providing a principled justification for this choice in the GP setting. Supported by our ablation in Appendix K, we use only $S = 5$ Monte Carlo samples for all guidance approximations, unless stated otherwise. Unless otherwise stated, kernel hyperparameters are estimated by maximising the marginal likelihood on the linear-Gaussian component $p(\boldsymbol{f}_0 \mid \mathcal{D})$ before applying our ODE sampler to the full predictive distribution $p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C})$. Performance is evaluated on held-out test data using root mean squared error (RMSE) and negative log predictive density (NLPD), estimated empirically from 100 samples. Reported timings include both hyperparameter fitting and sampling, averaged over 100 independent runs. All experiments used a Nvidia A4500 GPU workstation except the one-dimensional monotonic experiment which used an Apple M2 Pro processor with 32GB RAM to allow fair comparisons with competitor approaches. Code is available at `github.com/FlowInference/flow-inference`, with all additional implementation details provided in Appendices G-J.

6.1 Monotone and bounded regression: sanity check on a well-studied problem

Figure 3 validated FLOWGP on a well-studied benchmark, demonstrating that our general framework can match the performance of custom problem-specific approaches. We reproduce the case study of [47]: recovering $f(x) = \frac{1}{3}\left[\tan^{-1}(20x - 10) - \tan^{-1}(-10)\right]$ from seven observations under monotonicity and the bound $0 < f(x) < \frac{1}{3}\log(30x + 1) + 0.1 = u(x)$. Both constraints are handled as inequality conditions via the probit relaxation of Section 5.2: monotonicity is encoded as a lower bound on finite differences of $\boldsymbol{f}_0$, and boundedness as pointwise upper and lower bounds. Figure 4 in Appendix G shows that FLOWGP closely matches or outperforms bespoke methods from the statistics literature [48, 49] that are specifically engineered for this constraint structure, whilst avoiding the computational overhead of virtual observation approaches [47, 50, 51, 52].

Model	RMSE	NLPD	Time
PHYSS	**0.05**	-0.38	$1{\cdot}10^2$
AUTOIP	**0.05**	-0.08	$1{\cdot}10^2$
FLOW	**0.05 (0.00)**	-0.68 (0.10)	$\mathbf{4{\cdot}10^0}$
FLOW_W	**0.05 (0.00)**	**-1.05 (0.15)**	$\mathbf{4{\cdot}10^0}$

Model	RMSE	NLPD	Time
PHYSS	0.17	1.69	$5{\cdot}10^3$
AUTOIP	0.17	-0.29	$1{\cdot}10^4$
FLOW	0.38 (1.39)	3.20 (0.05)	$\mathbf{4{\cdot}10^0}$
FLOW_W	**0.13(0.00)**	**-0.83(0.02)**	$\mathbf{4{\cdot}10^0}$

Table 1: FLOWGP matches existing physics-obeying approaches in accuracy at orders-of-magnitude lower cost across physics benchmarks, with whitening ($_W$) consistently improving uncertainty quantification. **Left:** Non-linear pendulum ($25\times$ lower cost). **Right:** Allen–Cahn equation ($2{,}500\times$ lower cost). Timings in seconds; results to 2 d.p. with one standard deviation (in parentheses) and best-performing score in **bold**, each based on 100 independent runs, each producing 100 samples in a single batched computation. Baseline results are taken from the respective publications, which did not report confidence intervals but used comparable hardware (Nvidia Titan RTX vs. our Nvidia A4500).

6.2 Physics-obeying GPs: non-linear constraints without bespoke kernels or inference

We demonstrate FLOWGP on three physics benchmarks — the non-linear pendulum, Allen–Cahn, and Burgers' equations — each handled by specifying only the likelihood $p(\mathcal{C} \mid \boldsymbol{f}_0)$ via ODE and boundary condition residuals. Figure 1 shows physically consistent samples across all three, and Table 1 confirms competitive accuracy at orders-of-magnitude lower cost with improved uncertainty quantification over the first two benchmarks. See Appendix H.4 for qualitatively similar results on Burger's equations and an additional comparison with non-probabilistic kernel method of [38]. Physics constraints are enforced via finite differences at the discretisation grid; finer grids or spectral differentiation are straightforward substitutes requiring no changes to the sampling algorithm. We note that hyperparameter fitting procedures differ between methods: FLOWGP fits by maximising the marginal likelihood, whilst PHYSS and AUTOIP optimise within their respective inference frameworks. This may partially confound accuracy comparisons, e.g. the PDE constraint may strongly govern smoothness and lengthscale.

6.3 Bayesian Optimisation: FLOWGP supports well-calibrated decision making

GPs are widely used for decision-making under uncertainty, with Bayesian optimisation (BO) being the most prominent example; see [53] for a discussion. Certain advanced BO settings benefit from structural constraints: [54] show that for Bayesian Optimisation with Preference Exploration (BOPE), replacing the standard preference-learning GP (i.e. a probit-likelihood GP that learns via pairwise comparison data using a Laplace approximation) [55] with a carefully engineered Monotonic Neural Network Ensemble (MoNNE) yields state-of-the-art performance. We consider exactly this setting, but instead of a bespoke architecture, we simply apply FLOWGP with monotonicity as a constraint on the standard preference-learning GP. As can be seen in Figure 5 of Appendix I, we found FLOWGP to significantly improve GP performance across all six benchmark functions found in [54], often matching MoNNE performance. FLOWGP thus provides a principled and general way to incorporate structural knowledge into BO, where well-calibrated uncertainty is critical and bespoke architectures are typically required.

6.4 Analytical test: sensitivity to sharp likelihoods

Our penultimate experiment tests FLOWGP where the ground truth is known: conditioning a single draw from a 1D RBF GP prior on n noiseless observations at random locations. We run FLOWGP (1000 ODE steps, $S = 5$) with point-wise Gaussian-likelihood guidance at noise scale σ, and report the 2-Wasserstein distance between a Gaussian fitted to 1000 guided samples and the true posterior (Table 2; mean and 95% CI over 10 runs). Accuracy is not monotonic in σ: the distance first falls as the guidance becomes more informative, then rises for $\sigma \leq 10^{-6}$. This matches Theorem E.1, whose discretisation constant grows with the likelihood curvature $1/\sigma^2$, so at a fixed step size discretisation error eventually dominates. At these scales the noise variance is also below the nugget typically used to stabilise GP computations, so some numerical difficulty is expected.

Table 2: W_2 distance between FlowGP samples and the true GP posterior (mean $\pm$ 95% CI over 10 runs).

σ	$M = 2$	$M = 10$	$M = 40$
10^{-2}	0.16 ± 0.01	0.57 ± 0.01	0.27 ± 0.01
10^{-3}	0.17 ± 0.01	0.28 ± 0.01	0.05 ± 0.01
10^{-4}	0.18 ± 0.02	0.15 ± 0.01	0.05 ± 0.01
10^{-5}	0.16 ± 0.02	0.12 ± 0.01	0.05 ± 0.01
10^{-6}	0.47 ± 0.01	0.63 ± 0.01	0.67 ± 0.01
10^{-7}	0.49 ± 0.01	0.64 ± 0.01	0.68 ± 0.01

6.5 LLM-guided Gaussian processes: conditioning on anything, including text

Our final experiment demonstrates conditioning on natural language, serving as a proof of concept highlighting a new frontier enabled by our framework. We demonstrate conditioning on natural language via a product-of-experts construction, defining the target as $\pi(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \propto p(\boldsymbol{f}_0 \mid \mathcal{D}) q(\boldsymbol{f}_0 \mid \mathcal{C})$, where $q(\boldsymbol{f}_0 \mid \mathcal{C})$ is an LLM-derived density (Qwen3.5 [56]) over functions conditioned on text, as used previously to guide neural diffusion processes [57]. The score decomposes as $\nabla_{\boldsymbol{f}_t} \log \pi(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}) + \nabla_{\boldsymbol{f}_t} \log \tilde{q}(\boldsymbol{f}_t \mid \mathcal{C})$, where $\tilde{q}(\boldsymbol{f}_t \mid \mathcal{C}) := \int q(\boldsymbol{f}_0 \mid \mathcal{C}) p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) \mathrm{d}\boldsymbol{f}_0$ is the Gaussian-smoothed LLM density. Here, the guidance term is structurally identical to Section 4.2 and our Monte Carlo approximation applies directly. We adopt the product-of-experts to avoid backpropagation through the LLM, though direct conditioning is feasible given sufficient compute. Figure 1 shows that FLOWGP's samples are simultaneously prior-consistent and faithful to semantic descriptions. It takes less than 2 seconds to generate 100 samples for each prompt. Full text prompts and GP priors are given in Appendix J. We anticipate that as LLMs become richer sources of "qualitative data", FLOWGP provides a principled mechanism for incorporating that knowledge into GP models.

7 Limitations and future work

We have established a plug-and-play GP inference scheme supporting arbitrary likelihoods by leveraging an equivalence between Gaussian processes and linear diffusion models, with predictive sampling implemented as a guided ODE. Three key limitations of FLOWGP remain.

Scalability and discretisation. Cubic scaling in m makes large-scale problems intractable, and all theoretical results are stated for the finite-dimensional discretisation rather than the function-space limit; a rigorous infinite-dimensional treatment, together with integration with sparse GP approximations and spectral differentiation schemes, remains open. A version of FLOWGP that uses a random Fourier feature decomposition instead of a fine discretization of $\mathcal{X}$ can be found in [58]. It has also been shown that neural score-approximations trained over GP priors can largely ameliorate such computation cost [59].

Challenging likelihoods. Theoretical guarantees are strong only for log-concave likelihoods; non-log-concave settings admit only worst-case exponential bounds. Nevertheless, FLOWGP performs well empirically across all settings, including those where theory is weakest. We conjecture that Gaussian smoothing induced by the noising process regularises non-log-concave likelihoods in practice. Importance weight collapse remains a plausible concern in high dimensions. Understanding the precise conditions under which these failure modes arise, and developing adaptive remedies is an important direction for future work. Alternative parametrisations of the diffusion model may also yield advantageous performance [60].

Hyperparameter misspecification. Kernel hyperparameters are estimated from $p(\boldsymbol{f}_0 \mid \mathcal{D})$ alone, introducing a systematic bias when $\mathcal{C}$ is highly informative. A natural remedy is to optimise hyperparameters jointly over $p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C})$, perhaps by backpropagating through the ODE solver, though this introduces additional computational cost. Adaptive remedies for prior–constraint incompatibility, such as re-initialising hyperparameters when the guidance term dominates, are also natural extensions.

Acknowledgments and Disclosure of Funding

HM and CN were supported by Lancaster University's Mathematics for AI in Real-world Systems E3 grant. HM was also supported by a UKRI-funded Future Leaders Fellowship, and Schmidt Sciences. CD & PH acknowledge financial support by the European Research Council through ERC CoG Action 101123955 ANUBIS; the DFG Cluster of Excellence "Machine Learning - New Perspectives for Science", EXC 2064/2; the DFG SPP 2298 (Project HE 7114/6-1), and the Carl Zeiss Foundation project "Certification and Foundations of Safe Machine Learning Systems in Healthcare", as well as funds from the Ministry of Science, Research and Arts of the State of Baden-Württemberg. CN was supported by EPSRC grants EP/V022636/1 and EP/Y028783/1. The research was partially enabled by the ARC ITRH for Transforming Energy Infrastructure through Digital Engineering (TIDE; Grant No. IH200100009). TC was supported by the UKRI Centre for Doctoral Training in Applications of Artificial Intelligence to the study of Environmental Risks [EP/S022961/1].

References

[1] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2006.

[2] Håvard Rue and Sara Martino. Approximate Bayesian inference for hierarchical Gaussian Markov random field models. *Journal of Statistical Planning and Inference*, 137(10):3177–3192, 2007.

[3] Botond Cseke and Tom Heskes. Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research*, 12:417–454, 2011.

[4] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

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

[7] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Michael Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

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

[10] Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

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

[12] Sifan Wang, Zehao Dou, Siming Shan, Tong-Rui Liu, and Lu Lu. FunDiff: Diffusion models over function spaces for physics-informed generative modeling. *Nature Communications*, 17(5749), 2026.

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

[14] Marvin Pförtner, Ingo Steinwart, Philipp Hennig, and Jonathan Wenger. Physics-informed Gaussian process regression generalizes linear PDE solvers. *arXiv preprint arXiv:2212.12474*, 2022.

[15] Thomas Mejer Hansen, Andre G Journel, Albert Tarantola, and Klaus Mosegaard. Linear inverse Gaussian theory and geostatistics. *Geophysics*, 71(6):R101–R111, 2006.

[16] Ercan Solak, Roderick Murray-Smith, WE Leithead, Douglas Leith, and Carl Rasmussen. Derivative observations in Gaussian process models of dynamic systems. *Advances in Neural Information Processing Systems*, 15, 2002.

[17] Aurelien Luigi Serge Ponte, Lachlan C Astfalck, Matthew D Rayson, Andrew P Zulberti, and Nicole L Jones. Inferring flow energy, space scales, and timescales: freely drifting vs. fixed-point observations. *Nonlinear Processes in Geophysics*, 31(4):571–586, 2024.

[18] Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

[19] François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, and Dino Sejdinovic. Probabilistic integration. *Statistical Science*, 34(1):1–22, 2019.

[20] Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained Gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

[21] Peter Whittle. On stationary processes in the plane. *Biometrika*, pages 434–449, 1954.

[22] 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.

[23] Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

[24] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International conference on machine learning*, pages 10292–10302. PMLR, 2020.

[25] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[26] Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

[27] Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784. PMLR, 2015.

[28] Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In *International Conference on Machine Learning*, pages 2529–2538. PMLR, 2016.

[29] Jonathan Wenger, Kaiwen Wu, Philipp Hennig, Jacob R Gardner, Geoff Pleiss, and John P Cunningham. Computation-aware Gaussian processes: model selection and linear-time inference. *Advances in Neural Information Processing Systems*, 37:31316–31349, 2024.

[30] David Ginsbourger, Xavier Bay, Olivier Roustant, and Laurent Carraro. Argumentwise invariant kernels for the approximation of invariant functions. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 501–527, 2012.

[31] Mauricio A Alvarez, 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.

[32] David Ginsbourger, Olivier Roustant, and Nicolas Durrande. On degeneracy and invariances of random fields paths with applications in gaussian process modelling. *Journal of statistical planning and inference*, 170:117–128, 2016.

[33] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[34] Andreas Besginow and Markus Lange-Hegermann. Constraining Gaussian processes to systems of linear ordinary differential equations. *Advances in Neural Information Processing Systems*, 35:29386–29399, 2022.

[35] Tim Steinert, David Ginsbourger, August Lykke-Møller, Ove Christiansen, and Henry Moss. Integration-free kernels for equivariant gaussian process modelling. In *ICML*, 2025.

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

[37] Da Long, Zheng Wang, Aditi Krishnapriyan, Robert Kirby, Shandian Zhe, and Michael Mahoney. AutoIP: A united framework to integrate physics into Gaussian processes. In *International Conference on Machine Learning*, pages 14210–14222. PMLR, 2022.

[38] 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.

[39] Michael Schober, Simo Särkkä, and Philipp Hennig. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 2019.

[40] Filip Tronarp, Hans Kersting, Simo Särkkä, and Philipp Hennig. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 2019.

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

[42] Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

[43] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. The International Conference on Learning Representations, 2023.

[44] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations, ICLR 2024*. The International Conference on Learning Representations, 2024.

[45] Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

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

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

[49] Lachlan Astfalck, Deborshee Sen, Sayan Patra, Edward Cripps, and David Dunson. Posterior projection for inference in constrained spaces. *arXiv e-prints*, pages arXiv–1812, 2025.

[50] Fangpo Wang and Alan E Gelfand. Modeling space and space-time directional data using projected Gaussian processes. *Journal of the American Statistical Association*, 109(508):1565–1580, 2014.

[51] Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 645–652. JMLR Workshop and Conference Proceedings, 2010.

[52] Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. In *Annales de la Faculté des Sciences de Toulouse: Mathématiques*, pages 529–555, 2012.

[53] Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

[54] Hanyang Wang, Juergen Branke, and Matthias Poloczek. Bayesian optimization with preference exploration using a monotonic neural network ensemble. *Advances in Neural Information Processing Systems*, 2025.

[55] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *International Conference on Machine Learning*, 2005.

[56] Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

[57] Felix Biggs and Samuel Willis. LLM flow processes for text-conditioned regression. *arXiv preprint arXiv:2601.06147*, 2026.

[58] Thomas Cowperthwaite, Louis Sharrock, Lachlan Astfalck, and Henry Moss. Feature space adaptation for effortless Gaussian process flows. *arXiv preprint arXiv:2610.11459*, 2026.

[59] Andrew Zammit-Mangion, Lachlan Astfalck, Raphaël Huser, Christopher Nemeth, and Henry B Moss. Spatial prediction with guided diffusion. *arXiv preprint arXiv:2610.12967*, 2026.

[60] Louis Sharrock, Lachlan Astfalck, and Henry Moss. Latentflow: A general framework for conditioning stochastic processes. *arXiv preprint arXiv:2607.12922*, 2026.

[61] Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. *Statistical Optimal Transport*. Springer, Cham, Switzerland, 2025.

[62] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[63] Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, Germany, 1993.

[64] András Prékopa. On logarithmic concave measures and functions. *Acta Sci. Math.*, 34:335, 1973.

[65] Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: a review. *Statistics Surveys*, 8:45, 2014.

[66] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[67] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*, volume 10. Cambridge University Press, Cambridge, UK, 2019.

[68] Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. Preference exploration for efficient Bayesian optimization with multiple outcomes. In *International Conference on Artificial Intelligence and Statistics*, 2022.

[69] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

[70] Carl Hvarfner, Erik O Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *International Conference on Machine Learning*, 2024.

A Derivation of our flow's marginal and joint distributions

Under the prior $\boldsymbol{f}_0 \sim \mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$ and the corruption model (8), the pair $(\boldsymbol{f}_0, \boldsymbol{f}_t)$ is jointly Gaussian. Writing

$$\boldsymbol{f}_t = \alpha(t)\boldsymbol{f}_0 + \sqrt{1-\alpha^2(t)}\boldsymbol{z}, \qquad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m), \qquad \boldsymbol{z} \perp \boldsymbol{f}_0,$$

we obtain

$$\mathbb{E}[\boldsymbol{f}_t] = \alpha(t)\mathbf{m}_*, \qquad \mathrm{Cov}(\boldsymbol{f}_0, \boldsymbol{f}_t) = \alpha(t)\mathbf{K}_{**},$$

and

$$\mathrm{Cov}(\boldsymbol{f}_t) = \mathbf{A}(t) := \alpha^2(t)\mathbf{K}_{**} + (1-\alpha^2(t))\mathbf{I}_m. \tag{20}$$

Hence, the joint distribution is

$$\begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_* \\ \alpha(t)\mathbf{m}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{**} & \alpha(t)\mathbf{K}_{**} \\ \alpha(t)\mathbf{K}_{**} & \mathbf{A}(t) \end{bmatrix} \right).$$

Standard Gaussian conditioning then gives

$$\boldsymbol{f}_0 \mid \boldsymbol{f}_t \sim \mathcal{N}\big(\boldsymbol{\mu}_{0|t}, \boldsymbol{\Sigma}_{0|t}\big), \tag{21}$$

with

$$\boldsymbol{\mu}_{0|t} = \mathbf{m}_* + \alpha(t)\mathbf{K}_{**}\mathbf{A}(t)^{-1}\big(\boldsymbol{f}_t - \alpha(t)\mathbf{m}_*\big), \tag{22}$$

and

$$\boldsymbol{\Sigma}_{0|t} = \mathbf{K}_{**} - \alpha^2(t)\mathbf{K}_{**}\mathbf{A}(t)^{-1}\mathbf{K}_{**}. \tag{23}$$

In particular (as will be useful for Appendix C),

$$\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t] = \mathbf{m}_* + \alpha(t)\mathbf{K}_{**}\mathbf{A}(t)^{-1}\big(\boldsymbol{f}_t - \alpha(t)\mathbf{m}_*\big). \tag{24}$$

Note that conditioning on $\mathcal{D}$ simply replaces $\mathcal{N}(\mathbf{m}_*, \mathbf{K}_{**})$ by $\mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$, so $\boldsymbol{\mu}_{0|t}$ and $\boldsymbol{\Sigma}_{0|t}$ are the same expressions with $\mathbf{m}_{*|\boldsymbol{y}}$, $\mathbf{K}_{**|\boldsymbol{y}}$ and $\mathbf{A}_{|\boldsymbol{y}}(t)$ in place of $\mathbf{m}_*, \mathbf{K}_{**}$ and $\mathbf{A}(t)$.

Under whitening. In whitened space, the mean is $\mathbf{0}_m$ and the covariance matrix is $\mathbf{I}_m$, so we have the greatly simplified conditional distribution

$$\hat{\boldsymbol{f}}_0 \mid \hat{\boldsymbol{f}}_t \sim \mathcal{N}\big(\alpha(t)\hat{\boldsymbol{f}}_t, (1-\alpha^2(t))\mathbf{I}_m\big). \tag{25}$$

B Approximating the guidance term

For each ODE evaluation, we draw S samples $\boldsymbol{f}_0^{(i)} \sim p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}), i = 1, \ldots, S$, to obtain $p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) \approx \frac{1}{S}\sum_i p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$. We first take the gradient with respect to $\boldsymbol{f}_t$,

$$\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) \approx \nabla_{\boldsymbol{f}_t} \log \frac{1}{S}\sum_{i=1}^{S} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) = \sum_{i=1}^{S} \frac{\nabla_{\boldsymbol{f}_t} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})}{\sum_r p(\mathcal{C} | \boldsymbol{f}_0^{(r)})}, \tag{26}$$

and then apply the chain rule to each term $\nabla_{\boldsymbol{f}_t} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$,

$$\nabla_{\boldsymbol{f}_t} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) = \left[\frac{\partial \boldsymbol{f}_0^{(i)}}{\partial \boldsymbol{f}_t}\right]^T \nabla_{\boldsymbol{f}_0} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}). \tag{27}$$

Rewriting using the log-derivative trick, we obtain

$$\nabla_{\boldsymbol{f}_t} p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) = p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) \left[\frac{\partial \boldsymbol{f}_0^{(i)}}{\partial \boldsymbol{f}_t}\right]^T \nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}). \tag{28}$$

Substituting (28) into (26) yields

$$\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) \approx \sum_{i=1}^{S} \frac{p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) \left[\frac{\partial \boldsymbol{f}_0^{(i)}}{\partial \boldsymbol{f}_t}\right]^T \nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})}{\sum_r p(\mathcal{C} \mid \boldsymbol{f}_0^{(r)})}. \tag{29}$$

Introducing normalised importance weights $\bar{w}^{(i)} = p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) / \sum_r p(\mathcal{C} \mid \boldsymbol{f}_0^{(r)})$ and noting that, under our VP corruption kernel, the Jacobian is the symmetric matrix $\frac{\partial \boldsymbol{f}_0^{(i)}}{\partial \boldsymbol{f}_t} = \alpha(t)\mathbf{K}_{**}\mathbf{A}^{-1}(t)$, leads to our proposed guidance term approximation:

$$\boxed{\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) \approx \alpha(t)\mathbf{K}_{**|\boldsymbol{y}}\mathbf{A}_{|\boldsymbol{y}}^{-1}(t) \left[\sum_{i=1}^{S} \bar{w}^{(i)} \nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})\right].} \tag{30}$$

See Appendix F.3 for practical implementation details.

C An alternative guidance approximation using Fisher's Identity

For a gradient-free estimate of the score we may consider Fisher's Identity. First, consider the predictive distribution of $\boldsymbol{f}_t$. The derivative of this quantity is

$$\nabla_{\boldsymbol{f}_t} p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \int \nabla_{\boldsymbol{f}_t} p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \mathrm{d}\boldsymbol{f}_0.$$

Using

$$\nabla_{\boldsymbol{f}_t} p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) = p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) \nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \boldsymbol{f}_0),$$

we obtain

$$\nabla_{\boldsymbol{f}_t} p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \int p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) \nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \mathrm{d}\boldsymbol{f}_0.$$

Dividing by $p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C})$ on both sides and applying Bayes' Theorem gives

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = \mathbb{E}\left[\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) | \boldsymbol{f}_t, \mathcal{D}, \mathcal{C}\right]. \tag{31}$$

Under a variance preserving Gaussian corruption kernel we have that,

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \boldsymbol{f}_0) = -\frac{1}{1-\alpha^2(t)}\left(\boldsymbol{f}_t - \alpha(t)\boldsymbol{f}_0\right). \tag{32}$$

Substituting (32) into (31) therefore gives

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}, \mathcal{C}) = -\frac{1}{1-\alpha^2(t)}\left(\boldsymbol{f}_t - \alpha(t)\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}, \mathcal{C}]\right). \tag{33}$$

Similarly, in the case where we only condition on $\mathcal{D}$,

$$\nabla_{\boldsymbol{f}_t} \log p(\boldsymbol{f}_t \mid \mathcal{D}) = -\frac{1}{1-\alpha^2(t)}\left(\boldsymbol{f}_t - \alpha(t)\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}]\right). \tag{34}$$

From (15), we can subtract (34) from (33) to get the guidance identity:

$$\boxed{\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \boldsymbol{f}_t, \mathcal{D}) = \frac{\alpha(t)}{1-\alpha^2(t)}\left(\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}, \mathcal{C}] - \mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}]\right).} \tag{35}$$

The guidance is thus a scaled difference between the expected latent state $\boldsymbol{f}_0$ under full conditioning versus that conditional just on $\mathcal{D}$. Crucially, this decomposition separates tractable and intractable components, with the second term available in closed-form due to our knowledge of the conditional distribution under our noising process which, analogous to (9) , is given by

$$\begin{aligned}\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D} \sim \mathcal{N}\big(&\mathbf{m}_{*|\boldsymbol{y}} + \alpha(t)\mathbf{K}_{**|\boldsymbol{y}}\mathbf{A}_{|\boldsymbol{y}}(t)^{-1}\left(\boldsymbol{f}_t - \alpha(t)\mathbf{m}_{*|\boldsymbol{y}}\right), \\ &\mathbf{K}_{**|\boldsymbol{y}} - \alpha^2(t)\mathbf{K}_{**|\boldsymbol{y}}\mathbf{A}(t)^{-1}\mathbf{K}_{**|\boldsymbol{y}}\big).\end{aligned} \tag{36}$$

A Monte Carlo approximation of $\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}, \mathcal{C}]$: This conditional expectation is intractable for arbitrary non-linear or simulator-based likelihoods. However, we can exploit the fact that $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ in (36) is Gaussian, enabling a simple Monte Carlo approximation that requires the weaker assumption that our likelihood $p(\mathcal{C} \mid \boldsymbol{f}_0)$ can be evaluated point-wise. Specifically, for each ODE evaluation, we sample from the Gaussian distribution $\boldsymbol{f}_0^{(i)} \sim p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ for $i = 1, ..., S$, compute importance weights $w^{(i)} \propto p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$, and form the self-normalised approximation

$$\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D}, \mathcal{C}] \approx \sum_{i=1}^{S} \bar{w}^{(i)} \boldsymbol{f}_0^{(i)}, \qquad \bar{w}^{(i)} = \frac{w^{(i)}}{\sum_r w^{(r)}}. \tag{37}$$

This estimator has high variance when the conditioning event is sharp or the dimensionality is large, likely due to it computing a difference of conditional expectations rather than exploiting local shape information through a gradient. Indeed, in Appendix K we provide empirical evidence that this estimator requires orders of magnitude more samples than our gradient-based Monte Carlo estimator in Appendix B.

D Numerical Stability

We examine two ways to assess the discretisation error from the ODE solver. The first is via *stiffness*, which provides an notion of maximal instantaneous change in the solution of the linear system. This is an important notion for solvers with fixed discretisation. We will show that this is linked to the condition number of a GP. The next is via *transport*, which provides a measure of change from the initial to final states. This provides a richer notion of solver error as it characterises the whole path.

D.1 Stiffness

For prior and linear conditional sampling, the numerical stability of the ODE solution can be characterised through the *stiffness* of the underlying dynamical system. Loosely, stiffness provides a measure of the step size required for stable numerical integration: stiffer systems require relatively small time steps, and vice-versa. Consider a linear ODE of the form $\frac{d\boldsymbol{f}_t}{dt} = -\mathbf{D}(t)\, \boldsymbol{f}_t$. For general $\mathbf{D}(t)$, a common definition of stiffness is

$$\text{stiffness} = \max_{t \in [0,1]} \left\{ \frac{\sigma_{\max}(\mathbf{D}(t))}{\sigma_{\min}(\mathbf{D}(t))} \right\} \tag{38}$$

where $\sigma_{\max}(\mathbf{D}(t))$ and $\sigma_{\min}(\mathbf{D}(t))$ are the maximum and minimum singular values of $\mathbf{D}(t)$, respectively. In the traditional setting of diffusion models, the matrix $\mathbf{D}(t)$ depends on the unknown score function and is therefore not available in closed form. In contrast, in our GP setting where $p(\boldsymbol{f}_0)$ is specified by a GP prior, (5) admits an explicit linear representation. Below, we provide Theorem D.1 that applies uniformly to both prior sampling with $\mathbf{K}_{**}$ (from 5) and linear-conditional sampling with $\mathbf{K}_{**|\boldsymbol{y}}$ (from 11). In what follows, $\mathbf{K}$ denotes either case, as appropriate.

Theorem D.1. *As $n \to \infty$, define a sequence $\{\boldsymbol{K}_n\}_{n \in \mathbb{Z}^+}$ with corresponding sequence of condition numbers $\kappa(\boldsymbol{K}_n) = \lambda_{n,\max}/\lambda_{n,\min}$. Assume the regime where maximum eigenvalues $\lambda_{n,\max} < 1$ (attainable by rescaling $\boldsymbol{K}_n$; see the proof) are bounded in n (i.e. $\sup_n \lambda_{n,\max} < 1$) and the sequence of minimum eigenvalues $\lambda_{n,\min} \to 0$. Define the corresponding stiffness sequence S_n induced by $\boldsymbol{K}_n$ in (5). Then*

$$S_n = \kappa(\mathbf{K}_n)\, \frac{1 - \lambda_{n,\min}}{1 - \lambda_{n,\max}},$$

and consequently $S_n \asymp \kappa(\boldsymbol{K}_n)$ has double-sided asymptotic equivalence as $n \to \infty$.

Proof. Since $\kappa(\mathbf{K}_n)$ is invariant to positive scalar rescaling, we run the ODE on the rescaled problem $\mathbf{K}_n/\sigma_n^2$ with $\sigma_n^2 = c\, \lambda_{n,\max}$ for a fixed $c > 1$, and recover samples of $\boldsymbol{f}_0$ by multiplying by σ_n; this gives $\sup_n \lambda_{n,\max} = 1/c < 1$, and below $\mathbf{K}_n$ denotes the rescaled matrix. The GP probability-flow ODE has negative Jacobian

$$\mathbf{D}(t) = \frac{1}{2}\beta(t)(\mathbf{I}_m - \mathbf{A}(t)^{-1}) \tag{39}$$

where $\mathbf{A}(t) = \alpha^2(t)\mathbf{K} + (1-\alpha^2(t))\mathbf{I}_n$. Denote here the ordered set of eigenvalues of $\mathbf{A}(t)$ as $\lambda_1(t) \geq \cdots \geq \lambda_n(t) > 0$. From (39), the eigenvalues of $\mathbf{D}(t)$ are given as

$$\mu_i(t) = \frac{1}{2}\beta(t)(1-\lambda_i^{-1}(t)),$$

and since $\mathbf{D}(t)$ is symmetric, its singular values are

$$\sigma_i(t) = |\mu_i(t)| = \frac{1}{2}\beta(t)\left|1-\lambda_i^{-1}(t)\right|.$$

Thus, the stiffness at time t is

$$\text{stiffness}(t) = \frac{\sigma_{\max}(t)}{\sigma_{\min}(t)} = \frac{1-\lambda_n(t)^{-1}}{1-\lambda_1(t)^{-1}} = \kappa(\mathbf{A}(t)) \cdot \frac{1-\lambda_n(t)}{1-\lambda_1(t)}.$$

We further know that $\lambda_i(t) = \alpha^2(t)\lambda_i(0) + 1 - \alpha^2(t)$ for some monotonically decreasing $\alpha^2(t)$ with $\alpha^2(0) = 1$ and $\alpha^2(1) > 0$. Set $a(t) \coloneqq \alpha^2(t) \in (0,1]$ and write $\lambda_i(t) = 1 + a(t)(\lambda_i(0) - 1)$. Further define

$$S(a) = \frac{\lambda_1(a)}{\lambda_n(a)} \cdot \frac{1-\lambda_n(a)}{1-\lambda_1(a)}, \qquad \lambda_i(a) = 1 + a(\lambda_i(0) - 1).$$

Define $u_i \coloneqq \lambda_i(0) - 1$, so that $u_i < 0$ under the assumption $\lambda_i(0) < 1$, and

$$\lambda_i(a) = 1 + au_i, \qquad 1 - \lambda_i(a) = -au_i.$$

Hence,

$$S(a) = \frac{1+au_1}{1+au_n} \cdot \frac{-au_n}{-au_1} = \frac{u_n}{u_1} \cdot \frac{1+au_1}{1+au_n}.$$

Taking logarithms,

$$\log S(a) = \log\left(\frac{u_n}{u_1}\right) + \log(1+au_1) - \log(1+au_n),$$

and therefore

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}a}\log S(a) &= \frac{u_1}{1+au_1} - \frac{u_n}{1+au_n} \\ &= \frac{u_1(1+au_n) - u_n(1+au_1)}{(1+au_1)(1+au_n)} \\ &= \frac{u_1 - u_n}{(1+au_1)(1+au_n)}.\end{aligned}$$

Since $u_1 - u_n = \lambda_1(0) - \lambda_n(0) \geq 0$ and $\lambda_i(a) = 1 + au_i > 0$ for all $a \in (0,1]$, the denominator is strictly positive. Hence

$$\frac{\mathrm{d}}{\mathrm{d}a}\log S(a) \geq 0,$$

so $S(a)$ is non-decreasing in a.

As $a(t) = \alpha^2(t)$ is decreasing in t, it follows that $S(t)$ is strictly decreasing in t. Therefore the maximal stiffness is attained at $t = 0$, where $\mathbf{A}(0) = \mathbf{K}_n$, giving

$$S_n = \max_{t\in[0,1]} \text{stiffness}(t) = \text{stiffness}(0) = \kappa(\mathbf{K}_n)\,\frac{1-\lambda_{n,\min}}{1-\lambda_{n,\max}}.$$

Finally, since $\sup_n \lambda_{n,\max} < 1$ and $\lambda_{n,\min} \to 0$, the factor

$$\frac{1-\lambda_{n,\min}}{1-\lambda_{n,\max}}$$

is bounded above and below by positive constants independent of n. Hence $S_n \asymp \kappa(\mathbf{K}_n)$ which establishes the claimed double-sided asymptotic equivalence. □

Theorem D.1 makes clear that the diffusion formulation does not circumvent the fundamental numerical difficulties associated with ill-conditioned GPs. In the prior and linearly conditioned settings, where Theorem D.1 is valid, stiffness is governed by the same spectral quantities that control the condition number of the Gram matrix. In this sense, the diffusion framework provides an alternative sampling mechanism, but it does not eliminate the intrinsic spectral ill-conditioning of the underlying GP. When the conditioning information is no longer linear and Gaussian, an explicit expression for $\mathbf{D}(t)$, and hence for stiffness, is typically unavailable. Nevertheless, the notion of stiffness remains well defined. Intuitively, increasingly informative or highly nonlinear conditioning introduces strong anisotropy and local curvature in the conditional score, aligning with the asymptotic regime studied.

D.2 Transport

Stiffness and transport quantify fundamentally different aspects of the generative flow. Stiffness is a local, trajectory-level property: it reflects sensitivity of the ODE to perturbations. Transport, by contrast, is a measure-level property: it quantifies how much probability mass is displaced to transform $p(\boldsymbol{f}_1)$ into $p(\boldsymbol{f}_0)$. In effect, stiffness characterises numerical sensitivity of sample paths, but it does not quantify how much mass is rearranged to transform the base distribution into the target, or vice versa. This is instead governed by the geometry of the distributional path and necessitates a measure-level perspective. To make this precise, we're required to explicitly index the time-indexed family of distributions of the process. For the stochastic process $\{\boldsymbol{f}_t\}_{t\in[0,1]}$, denote by p_t the density of $\boldsymbol{f}_t$ at time t. In particular, p_0 and p_1 correspond to the target and base distributions, respectively. We will therefore work with the curve $\{p_t\}_{t\in[0,1]}$ in the space of probability measures.

We quantify the cost of transporting a distributional path $\{p_t\}_{t\in[0,1]}$ by its kinetic action. Given a velocity field $v(\cdot,t)$, e.g. (5) or (12), define

$$\mathcal{A} := \frac{1}{2}\int_0^1 \mathbb{E}\big[\|v(\boldsymbol{f}_t,t)\|^2\big]\,dt = \frac{1}{2}\int_0^1 \int \|v(\boldsymbol{f},t)\|^2\, p_t(\boldsymbol{f})\,d\boldsymbol{f}\,dt. \tag{40}$$

In general, [61, Chapter 5] shows that the pair $\{p_t, v_t\}_{t\in[0,1]}$ is admissible if it satisfies the continuity equation

$$\partial_t p_t(\boldsymbol{f}) + \nabla_{\boldsymbol{f}}\cdot\big(p_t(\boldsymbol{f})\,v(\boldsymbol{f},t)\big) = 0. \tag{41}$$

A given path $\{p_t\}_{t\in[0,1]}$ may admit multiple velocity fields satisfying (41), and these may yield different values of the action (40). The Benamou–Brenier theorem [62] identifies the minimum possible action.

Theorem D.2 (Benamou–Brenier)**.** *Let $p_0, p_1 \in \mathcal{P}_{2,\mathrm{ac}}(\mathbb{R}^d)$. Then*

$$W_2^2(p_0,p_1) = \inf_{\{p_t,v_t\}}\left\{\int_0^1\int \|v(\boldsymbol{f},t)\|^2\,p_t(\boldsymbol{f})\,d\boldsymbol{f}\,dt \;\middle|\; \partial_t p_t + \nabla\cdot(p_t v_t) = 0\right\}$$

where $W_2^2(p_0,p_1)$ is the squared Wasserstein–2 distance between p_0 and p_1. In particular, every admissible pair $\{p_t, v_t\}$ satisfies $W_2^2(p_0,p_1) \le 2\mathcal{A}$.

In particular, in the Gaussian setting considered herein we have $p_t = \mathcal{N}(0,\mathbf{A}(t))$ and linear velocity field $v(\boldsymbol{f},t) = -\mathbf{D}(t)\,\boldsymbol{f}$. This leads to Theorem D.3 that upper bounds the Wasserstein-2 distance.

Theorem D.3. *Suppose $p_t = \mathcal{N}(0,\boldsymbol{A}(t))$, where $\boldsymbol{A}(t) \in \mathbb{R}^{n\times n}$ is symmetric positive definite for each $t\in[0,1]$, and suppose the probability flow velocity $v(\boldsymbol{f},t) = -\boldsymbol{D}(t)\,\boldsymbol{f}$, where $\boldsymbol{D}(t) = \frac{1}{2}\beta(t)\big(I - \boldsymbol{A}(t)^{-1}\big)$. Then*

$$W_2^2(p_0,p_1) \le \frac{1}{4}\int_0^1 \beta(t)^2 \sum_{i=1}^n \frac{(\lambda_i(t)-1)^2}{\lambda_i(t)}\,dt,$$

where $\lambda_1(t),\dots,\lambda_n(t)$ are the eigenvalues of $\boldsymbol{A}(t)$.

Proof. Since $\boldsymbol{f}_t \sim \mathcal{N}(0,\mathbf{A}(t))$ and $v(\boldsymbol{f}_t,t) = -\mathbf{D}(t)\boldsymbol{f}_t$,

$$\mathbb{E}\big[\|v(\boldsymbol{f}_t,t)\|^2\big] = \mathbb{E}[\boldsymbol{f}_t^{\intercal}\mathbf{D}(t)^{\intercal}\mathbf{D}(t)\,\boldsymbol{f}_t] = \mathrm{Tr}\big(\mathbf{D}(t)\,\mathbf{A}(t)\,\mathbf{D}(t)^{\intercal}\big),$$

and therefore

$$\mathcal{A} = \frac{1}{2}\int_0^1 \mathrm{Tr}\big(\mathbf{D}(t)\,\mathbf{A}(t)\,\mathbf{D}(t)^\top\big)\,dt.$$

Substituting $\mathbf{D}(t) = -\frac{1}{2}\beta(t)(\mathbf{I}_m - \mathbf{A}(t)^{-1})$ gives

$$\begin{aligned} W_2^2(p_0,p_1) \le 2\mathcal{A} &= \int_0^1 \mathrm{Tr}\big(\mathbf{D}(t)\,\mathbf{A}(t)\,\mathbf{D}(t)^\top\big)\,dt \\ &= \frac{1}{4}\int_0^1 \beta(t)^2\Big(\mathrm{Tr}(\mathbf{A}(t)) - 2n + \mathrm{Tr}(\mathbf{A}(t)^{-1})\Big)\,dt \\ &= \frac{1}{4}\int_0^1 \beta(t)^2 \sum_{i=1}^n \Big(\lambda_i(t) - 2 + \lambda_i(t)^{-1}\Big)\,dt \\ &= \frac{1}{4}\int_0^1 \beta(t)^2 \sum_{i=1}^n \frac{(\lambda_i(t)-1)^2}{\lambda_i(t)}\,dt. \end{aligned}$$

□

Finally, this leads to Corollary D.4 which minimises the upper bound.

Corollary D.4. *Fix a measurable function $\beta : [0,1] \to \mathbb{R}$. The upper bound in Theorem D.3 is minimised when $\mathbf{A}(t) = \boldsymbol{I}$ for all $t \in [0,1]$. Moreover, the integrand admits the representation*

$$\sum_{i=1}^n \frac{(\lambda_i(t)-1)^2}{\lambda_i(t)} \ge 0,$$

where $\lambda_1(t), \dots, \lambda_n(t)$ are the eigenvalues of $\mathbf{A}(t)$, with equality if and only if $\mathbf{A}(t) = \boldsymbol{I}$ for $\int_0^1 \beta(t)^2\,\mathrm{d}t > 0$. Consequently, the bound in Theorem D.3 vanishes when $\mathbf{A}(t) = \boldsymbol{I}$ for all $t \in [0,1]$.

Proof. For each eigenvalue $\lambda > 0$,

$$\lambda - 2 + \lambda^{-1} = \frac{(\lambda-1)^2}{\lambda} \ge 0,$$

with equality if and only if $\lambda = 1$. Summing over eigenvalues yields the result. □

E General perturbation result

FLOWGP samples by integrating (16) (or its whitened form (19)) on a finite grid, with the guidance term $g(t, \boldsymbol{f}) := \nabla_{\boldsymbol{f}} \log p(\mathcal{C} \mid \boldsymbol{f}, \mathcal{D})$ replaced by an approximation $\widehat{g}$, such as the Monte Carlo estimator of Section 4.2. There are therefore two sources of error: *(i)* the guidance approximation $\widehat{g} \neq g$, and *(ii)* the ODE solver's discretisation error. We give a deterministic, pathwise result that controls both errors once a realised approximate guidance field $\widehat{g}$ has been fixed. The result is stated on any truncated interval $[\tau, 1] \subset (0,1]$,[2] under a single regularity condition on the reversed-time guided drift

$$\boldsymbol{a}(r, \boldsymbol{f}) \;=\; -\boldsymbol{v}(\boldsymbol{f}, 1-r \mid \mathcal{D}) + \tfrac{1}{2}\beta(1-r)g(1-r, \boldsymbol{f}).$$

The condition is a one-sided Lipschitz bound [see e.g. 63], which strictly generalises global Lipschitz continuity and admits a constant of either sign.

Theorem E.1 (Pathwise stability of guided sampling)**.** *Fix $\tau \in (0,1)$ and assume that, on $[\tau,1] \times \mathbb{R}^m$:* (i) *$\beta$, $\boldsymbol{v}(\cdot,\cdot \mid \mathcal{D})$, g, and $\widehat{g}$ are continuous;* (ii) *the reversed-time exact drift $\boldsymbol{a}$ is one-sided Lipschitz, i.e. there exists $\eta_\tau \in \mathbb{R}$ such that*

$$\langle \boldsymbol{x} - \boldsymbol{y},\, \boldsymbol{a}(r,\boldsymbol{x}) - \boldsymbol{a}(r,\boldsymbol{y})\rangle \;\le\; \eta_\tau \|\boldsymbol{x} - \boldsymbol{y}\|^2 \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m,\, r \in [0, 1-\tau]; \tag{42}$$

(iii) *the realised guidance error is uniformly bounded,*

$$\varepsilon_\tau := \sup_{(t,\boldsymbol{f}) \in [\tau,1] \times \mathbb{R}^m} \|g(t,\boldsymbol{f}) - \widehat{g}(t,\boldsymbol{f})\| < \infty.$$

[2] Truncation away from $t = 0$ is needed because the bridge factor $\alpha(t)/(1-\alpha^2(t))$ appearing in the conditional score is unbounded as $t \downarrow 0$; in practice τ is taken small (e.g. $\tau = 10^{-3}$).

Let $\boldsymbol{f}_t$ denote the exact guided flow, let $\widehat{\boldsymbol{f}}_t$ denote the exact solution of the ODE with g replaced by $\widehat{g}$, and let $\widehat{\boldsymbol{f}}^h_{t_n}$ denote the numerical approximation to $\widehat{\boldsymbol{f}}_{t_n}$ obtained by a deterministic explicit solver of global order q (constant $C_{\mathrm{num},\tau}$, step size h). Then for every grid point $t_n \in [\tau, 1]$,

$$\|\boldsymbol{f}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\| \;\le\; \tfrac{1}{2} B_\tau \varepsilon_\tau \, \Psi_{\eta_\tau}(1 - t_n) \;+\; C_{\mathrm{num},\tau}\, h^q, \quad \textit{where} \quad \Psi_\eta(r) := \begin{cases} (e^{\eta r} - 1)/\eta, & \eta \neq 0, \\ r, & \eta = 0, \end{cases} \tag{43}$$

and $B_\tau := \sup_{t \in [\tau,1]} |\beta(t)|$.

The proof, together with detailed assumptions, is given below. We first discuss the implications of this result for FLOWGP.

Three regimes. The sign of the constant η_τ in (42) determines the qualitative behaviour of the bound (43). *(a) Worst case ($\eta_\tau > 0$):* Ψ_{η_τ} grows exponentially in $1 - t$, which is the standard conclusion under any global Lipschitz hypothesis on $\boldsymbol{v}$ and g, with $\eta_\tau = L_v + \frac{1}{2} B_\tau L_g$. *(b) Borderline ($\eta_\tau = 0$):* the perturbation accumulates linearly in $1 - t$. *(c) Contractive ($\eta_\tau < 0$):* Ψ_{η_τ} is bounded by $|\eta_\tau|^{-1}$, so the deterministic perturbation induced by the realised guidance field remains bounded rather than accumulating, and the total error reduces to $\frac{1}{2} B_\tau \varepsilon_\tau / |\eta_\tau| + C_{\mathrm{num},\tau} h^q$.

Which likelihoods sit where? When g is differentiable, $\nabla_{\boldsymbol{f}} g(t, \boldsymbol{f}) = \nabla^2_{\boldsymbol{f}} \log p(\mathcal{C} \mid \boldsymbol{f}, \mathcal{D})$, so the guidance contribution to η_τ is governed by the curvature of the smoothed log-likelihood. By Prékopa–Leindler [64], log-concavity of the base likelihood $p(\mathcal{C} \mid \boldsymbol{f}_0)$ is preserved by Gaussian smoothing; see also [65] for a review. Thus log-concave likelihoods give a non-expansive guidance contribution, while uniformly strongly log-concave likelihoods give a contractive contribution on the relevant region of state space. In whitened coordinates, where the Gaussian linear velocity vanishes, this curvature directly controls the sign of the one-sided Lipschitz constant. In non-whitened coordinates, the reversed Gaussian flow also contributes to η_τ.

Consequently, Gaussian observations and linear inverse problems provide the clearest examples of a strongly contractive guidance contribution. Probit and logistic classification, Poisson regression with canonical log link, and convex-set indicator constraints are log-concave but need not be uniformly strongly log-concave, so they more naturally correspond to borderline or weakly contractive regimes unless additional local curvature is available. Genuinely non-log-concave settings, including mixture observation models, nonlinear forward operators, and the LLM-based conditioning of Section 6.5, are covered only by the worst-case stability bound. Theorem E.1 therefore gives a unified deterministic perturbation statement, but it should not be read as proving contractivity for every likelihood used in Section 6.

E.1 Proof

This result proves the finite-horizon deterministic stability bound underlying the guidance error analysis of Section 6. We control the gap between the trajectories of the exact and approximate guided ODEs on $[\tau, 1]$ in terms of a uniform realised guidance error ε_τ and a single one-sided Lipschitz constant η_τ. The main result, Theorem E.1, specialises to three qualitative regimes according to the sign of η_τ: *worst case* ($\eta_\tau > 0$, exponential accumulation), *borderline* ($\eta_\tau = 0$, linear accumulation), and *contractive* ($\eta_\tau < 0$, the guidance perturbation remains bounded). Uniform strong log-concavity of the smoothed likelihood can yield the contractive regime, especially in whitened coordinates, but ordinary log-concavity alone should only be interpreted as a non-expansiveness statement for the guidance contribution.

Fix $\tau \in (0, 1)$. On the truncated interval $[\tau, 1]$, consider the exact and approximate guided ODEs

$$\frac{\mathrm{d}\boldsymbol{f}_t}{\mathrm{d}t} = \boldsymbol{b}(t, \boldsymbol{f}_t), \qquad \boldsymbol{b}(t, \boldsymbol{f}) := \boldsymbol{v}(\boldsymbol{f}, t \mid \mathcal{D}) - \tfrac{1}{2}\beta(t) g(t, \boldsymbol{f}), \tag{44}$$

$$\frac{\mathrm{d}\widehat{\boldsymbol{f}}_t}{\mathrm{d}t} = \widehat{\boldsymbol{b}}(t, \widehat{\boldsymbol{f}}_t), \qquad \widehat{\boldsymbol{b}}(t, \boldsymbol{f}) := \boldsymbol{v}(\boldsymbol{f}, t \mid \mathcal{D}) - \tfrac{1}{2}\beta(t) \widehat{g}(t, \boldsymbol{f}), \tag{45}$$

where

$$g(t, \boldsymbol{f}) := \nabla_{\boldsymbol{f}} \log p(\mathcal{C} \mid \boldsymbol{f}, \mathcal{D})$$

is the exact guidance field as derived in Section 4, and $\widehat{g}$ is its realised approximation. We truncate away from $t = 0$ because the bridge factors appearing in the conditional score, such as $\alpha(t)/(1 - \alpha^2(t))$, may become singular as $t \downarrow 0$.

It will be convenient to phrase regularity on the *reversed-time* drift, since sampling runs backward from $t = 1$, in line with the standard convention for probability-flow ODEs [41, 6]. Define

$$\boldsymbol{a}(r, \boldsymbol{f}) := -\boldsymbol{b}(1 - r, \boldsymbol{f}), \qquad \widehat{\boldsymbol{a}}(r, \boldsymbol{f}) := -\widehat{\boldsymbol{b}}(1 - r, \boldsymbol{f}), \qquad r \in [0, 1 - \tau], \tag{46}$$

so that

$$\boldsymbol{a}(r, \boldsymbol{f}) = -\boldsymbol{v}(\boldsymbol{f}, 1 - r \mid \mathcal{D}) + \tfrac{1}{2}\beta(1 - r)g(1 - r, \boldsymbol{f}),$$

and analogously for $\widehat{\boldsymbol{a}}$. Reversed time is set up so that integrating from $r = 0$ corresponds to integrating from $t = 1$ down to $t = \tau$.

Assumption E.2 (Uniform regularity of the guided drifts)**.** Fix $\tau \in (0, 1)$. Assume that:

1. $\beta : [\tau, 1] \to \mathbb{R}$ is continuous, with
$$B_\tau := \sup_{t \in [\tau, 1]} |\beta(t)| < \infty;$$

2. $\boldsymbol{v}(\cdot, \cdot \mid \mathcal{D})$, g, and $\widehat{g}$ are continuous on $[\tau, 1] \times \mathbb{R}^m$, so that the reversed-time drifts $\boldsymbol{a}$ and $\widehat{\boldsymbol{a}}$ defined in (46) are continuous on $[0, 1 - \tau] \times \mathbb{R}^m$;

3. the reversed-time exact drift $\boldsymbol{a}$ satisfies a one-sided Lipschitz condition in $\boldsymbol{f}$, uniformly in r: therefore there exists $\eta_\tau \in \mathbb{R}$ such that
$$\langle \boldsymbol{x} - \boldsymbol{y},\ \boldsymbol{a}(r, \boldsymbol{x}) - \boldsymbol{a}(r, \boldsymbol{y}) \rangle \le \eta_\tau \|\boldsymbol{x} - \boldsymbol{y}\|^2 \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m,\ r \in [0, 1 - \tau]; \tag{47}$$

4. the realised guidance approximation error is uniformly bounded on the relevant state-space region visited by the exact and approximate trajectories:
$$\varepsilon_\tau := \sup_{(t, \boldsymbol{f}) \in \mathcal{R}_\tau} \|g(t, \boldsymbol{f}) - \widehat{g}(t, \boldsymbol{f})\| < \infty,$$
where $\mathcal{R}_\tau \subseteq [\tau, 1] \times \mathbb{R}^m$ denotes any region containing both trajectories on the truncated interval.

The constant η_τ is allowed to be negative; this contractive case will be exploited in Corollary E.7 below. The uniform error condition should be interpreted as a deterministic perturbation assumption on the realised guidance field over the relevant state-space region visited by the sampler. When $\widehat{g}$ is computed by Monte Carlo, all bounds below are conditional on the realised Monte Carlo randomness unless additional stochastic uniform-error bounds are imposed.

For convenience, we define

$$C_\tau := \tfrac{1}{2} B_\tau \varepsilon_\tau, \qquad \Psi_\eta(r) := \begin{cases} \dfrac{e^{\eta r} - 1}{\eta}, & \eta \neq 0, \\ r, & \eta = 0, \end{cases} \quad r \ge 0. \tag{48}$$

Lemma E.3 (Properties of Ψ_η)**.** *The map $(\eta, r) \mapsto \Psi_\eta(r)$ defined by (48) satisfies:*

(i) $\Psi_\eta(0) = 0$ and $r \mapsto \Psi_\eta(r)$ is continuously differentiable with derivative $e^{\eta r} > 0$, and is therefore strictly increasing;

(ii) $(\eta, r) \mapsto \Psi_\eta(r)$ is jointly continuous on $\mathbb{R} \times [0, \infty)$; in particular $\Psi_\eta(r) \to r$ as $\eta \to 0$;

(iii) if $\eta < 0$, then
$$\Psi_\eta(r) = \frac{1 - e^{-|\eta| r}}{|\eta|} \le \frac{1}{|\eta|} \qquad \forall r \ge 0,$$
so Ψ_η is bounded uniformly in r.

Proof. Direct calculation. Joint continuity at $\eta = 0$ follows from the Taylor expansion $(e^{\eta r} - 1)/\eta = r + \frac{1}{2}\eta r^2 + O(\eta^2)$; the bound for $\eta < 0$ uses $1 - e^{-|\eta| r} \le 1$. □

Remark E.4 (How to verify Assumption E.2 (3)). The one-sided Lipschitz condition (47) can be checked through a hierarchy of increasingly favourable regimes.

(i) *Global Lipschitz drift.* If $\boldsymbol{b}$ is globally Lipschitz in $\boldsymbol{f}$, uniformly in $t \in [\tau, 1]$, with constant L_τ, then $\boldsymbol{a}(r, \cdot) = -\boldsymbol{b}(1 - r, \cdot)$ is also globally Lipschitz in $\boldsymbol{f}$ with the same constant. Hence, by Cauchy–Schwarz,

$$\langle \boldsymbol{x} - \boldsymbol{y},\ \boldsymbol{a}(r, \boldsymbol{x}) - \boldsymbol{a}(r, \boldsymbol{y}) \rangle \le \|\boldsymbol{x} - \boldsymbol{y}\|\, \|\boldsymbol{a}(r, \boldsymbol{x}) - \boldsymbol{a}(r, \boldsymbol{y})\| \le L_\tau \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Thus (47) holds with $\eta_\tau = L_\tau$. In particular, if $\boldsymbol{v}$ and g are globally Lipschitz in $\boldsymbol{f}$, uniformly in t, with constants L_v and L_g, respectively, then one can take

$$\eta_\tau = L_v + \tfrac{1}{2} B_\tau L_g.$$

(ii) *One-sided Lipschitz guidance.* Suppose $\beta(t) \ge 0$ on $[\tau, 1]$. Suppose also that g is one-sided Lipschitz with constant $\ell_\tau \in \mathbb{R}$, uniformly in t, namely

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

and that $-\boldsymbol{v}(\cdot, 1 - r \mid \mathcal{D})$ is one-sided Lipschitz with constant $\eta_{v,\tau}$, uniformly in r. Decomposing gives

$$\boldsymbol{a}(r, \boldsymbol{f}) = \underbrace{-\boldsymbol{v}(\boldsymbol{f}, 1 - r \mid \mathcal{D})}_{\text{Gaussian flow}} + \underbrace{\tfrac{1}{2}\beta(1 - r) g(1 - r, \boldsymbol{f})}_{\text{guidance term}},$$

and adding the two one-sided Lipschitz estimates, one obtains

$$\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{a}(r, \boldsymbol{x}) - \boldsymbol{a}(r, \boldsymbol{y}) \rangle \le \eta_{v,\tau} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \tfrac{1}{2}\beta(1 - r)\ell_\tau \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

Taking the supremum over $r \in [0, 1 - \tau]$ shows that (47) holds with

$$\eta_\tau = \eta_{v,\tau} + \tfrac{1}{2} \sup_{t \in [\tau, 1]} \{\beta(t)\ell_\tau\}.$$

In particular, if $\boldsymbol{v}$ is globally Lipschitz with constant L_v, one may take $\eta_{v,\tau} = L_v$. Writing

$$\overline{B}_\tau := \sup_{t \in [\tau, 1]} \beta(t), \qquad \underline{B}_\tau := \inf_{t \in [\tau, 1]} \beta(t),$$

the supremum is attained at the larger or smaller value of β depending on the sign of ℓ_τ:

$$\eta_\tau = L_v + \tfrac{1}{2}\overline{B}_\tau \ell_\tau \qquad \text{if } \ell_\tau \ge 0, \qquad \eta_\tau = L_v + \tfrac{1}{2}\underline{B}_\tau \ell_\tau \qquad \text{if } \ell_\tau < 0.$$

The latter case is the one in which contractivity of the guidance can improve the perturbation bound.

(iii) *Strongly monotone guidance.* Suppose $\boldsymbol{v} \equiv 0$, $\beta(t) \ge \underline{B}_\tau > 0$ on $[\tau, 1]$, and g satisfies the strong monotonicity condition

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

for some $\mu_\tau > 0$, uniformly in $t \in [\tau, 1]$. Then (47) holds with

$$\eta_\tau = -\tfrac{1}{2}\underline{B}_\tau \mu_\tau < 0,$$

and the perturbation bound saturates; see Corollary E.7.

Proposition E.5 (Uniform pathwise stability under approximate guidance)**.** *Let Assumption E.2 hold. Suppose that the exact and approximate reversed-time ODEs admit solutions on $[0, 1 - \tau]$ from a common initial condition, or equivalently that the original-time ODEs (44)–(45) admit solutions on $[\tau, 1]$ with common terminal condition $\boldsymbol{f}_1 = \widehat{\boldsymbol{f}}_1$. Then, for every $t \in [\tau, 1]$,*

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le C_\tau\, \Psi_{\eta_\tau}(1 - t), \tag{49}$$

which implies,

$$\sup_{t \in [\tau, 1]} \|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le C_\tau\, \Psi_{\eta_\tau}(1 - \tau). \tag{50}$$

Equivalently,

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le \begin{cases} \dfrac{B_\tau \varepsilon_\tau}{2\eta_\tau}\big(e^{\eta_\tau(1-t)} - 1\big), & \eta_\tau \ne 0, \\ \tfrac{1}{2} B_\tau \varepsilon_\tau\,(1 - t), & \eta_\tau = 0, \end{cases}$$

Proof. Switch to reversed time $r = 1 - t$, and define

$$\boldsymbol{y}_r := \boldsymbol{f}_{1-r}, \qquad \widehat{\boldsymbol{y}}_r := \widehat{\boldsymbol{f}}_{1-r}, \qquad \boldsymbol{e}_r := \boldsymbol{y}_r - \widehat{\boldsymbol{y}}_r, \qquad r \in [0, 1-\tau].$$

A change of variables in (44)–(45) gives

$$\frac{\mathrm{d}\boldsymbol{y}_r}{\mathrm{d}r} = \boldsymbol{a}(r, \boldsymbol{y}_r), \qquad \frac{\mathrm{d}\widehat{\boldsymbol{y}}_r}{\mathrm{d}r} = \widehat{\boldsymbol{a}}(r, \widehat{\boldsymbol{y}}_r), \qquad \boldsymbol{e}_0 = 0.$$

Hence

$$\boldsymbol{e}_r' = \boldsymbol{a}(r, \boldsymbol{y}_r) - \widehat{\boldsymbol{a}}(r, \widehat{\boldsymbol{y}}_r). \tag{51}$$

The map $r \mapsto \|\boldsymbol{e}_r\|^2$ is differentiable but $r \mapsto \|\boldsymbol{e}_r\|$ is not, near points where $\boldsymbol{e}_r = \boldsymbol{0}$. To circumvent this, we work with the strictly positive smooth surrogate

$$\phi_\varepsilon(r) := \sqrt{\|\boldsymbol{e}_r\|^2 + \varepsilon}, \qquad \varepsilon > 0,$$

which is continuously differentiable for every $\varepsilon > 0$ and satisfies $\phi_\varepsilon(0) = \sqrt{\varepsilon}$ (since $\boldsymbol{e}_0 = \boldsymbol{0}$), and therefore it follows that

$$\phi_\varepsilon'(r) = \frac{1}{2\phi_\varepsilon(r)} \frac{\mathrm{d}}{\mathrm{d}r} \|\boldsymbol{e}_r\|^2 = \frac{\langle \boldsymbol{e}_r, \boldsymbol{e}_r' \rangle}{\phi_\varepsilon(r)}. \tag{52}$$

We will derive a linear differential inequality for ϕ_ε, integrate it using the Grönwall inequality, and then let $\varepsilon \downarrow 0$.

We start by adding and subtracting $\boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r)$ in (51) to split $\boldsymbol{e}_r'$ into a contractive part and a drift error:

$$\boldsymbol{e}_r' = \underbrace{\boldsymbol{a}(r, \boldsymbol{y}_r) - \boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r)}_{\text{exact-drift difference}} + \underbrace{\boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r) - \widehat{\boldsymbol{a}}(r, \widehat{\boldsymbol{y}}_r)}_{\text{drift error}}.$$

Applying the one-sided Lipschitz assumption (47) to the first term with $\boldsymbol{x} = \boldsymbol{y}_r$ and $\boldsymbol{y} = \widehat{\boldsymbol{y}}_r$,

$$\langle \boldsymbol{e}_r, \boldsymbol{a}(r, \boldsymbol{y}_r) - \boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r) \rangle \le \eta_\tau \|\boldsymbol{e}_r\|^2. \tag{53}$$

The Gaussian flow $\boldsymbol{v}$ enters $\boldsymbol{b}$ and $\widehat{\boldsymbol{b}}$ identically and so cancels in $\boldsymbol{a} - \widehat{\boldsymbol{a}}$, meaning that for the second term we have,

$$\boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r) - \widehat{\boldsymbol{a}}(r, \widehat{\boldsymbol{y}}_r) = \tfrac{1}{2}\beta(1-r)\big(g(1-r, \widehat{\boldsymbol{y}}_r) - \widehat{g}(1-r, \widehat{\boldsymbol{y}}_r)\big).$$

By Cauchy–Schwarz and Assumption E.2 (1) and (4), we have

$$\big|\langle \boldsymbol{e}_r, \boldsymbol{a}(r, \widehat{\boldsymbol{y}}_r) - \widehat{\boldsymbol{a}}(r, \widehat{\boldsymbol{y}}_r) \rangle\big| \le \|\boldsymbol{e}_r\| \cdot \tfrac{1}{2} B_\tau \varepsilon_\tau = C_\tau \|\boldsymbol{e}_r\|. \tag{54}$$

Combining (53) and (54) gives,

$$\langle \boldsymbol{e}_r, \boldsymbol{e}_r' \rangle \le \eta_\tau \|\boldsymbol{e}_r\|^2 + C_\tau \|\boldsymbol{e}_r\|. \tag{55}$$

We divide by $\phi_\varepsilon(r) > 0$ and bound each term. For the drift error term,

$$C_\tau \frac{\|\boldsymbol{e}_r\|}{\phi_\varepsilon(r)} = C_\tau \frac{\|\boldsymbol{e}_r\|}{\sqrt{\|\boldsymbol{e}_r\|^2 + \varepsilon}} \le C_\tau,$$

since $\|\boldsymbol{e}_r\| \le \sqrt{\|\boldsymbol{e}_r\|^2 + \varepsilon}$. For the contraction term we use the identity $\|\boldsymbol{e}_r\|^2 = \phi_\varepsilon(r)^2 - \varepsilon$ to rewrite

$$\frac{\eta_\tau \|\boldsymbol{e}_r\|^2}{\phi_\varepsilon(r)} = \eta_\tau \phi_\varepsilon(r) - \frac{\eta_\tau \varepsilon}{\phi_\varepsilon(r)}. \tag{56}$$

The second term is bounded uniformly in r by

$$\left| \frac{\eta_\tau \varepsilon}{\phi_\varepsilon(r)} \right| \le \frac{|\eta_\tau| \varepsilon}{\sqrt{\varepsilon}} = |\eta_\tau| \sqrt{\varepsilon},$$

since $\phi_\varepsilon(r) \ge \sqrt{\varepsilon}$. The bound $|-\eta_\tau \varepsilon / \phi_\varepsilon| \le |\eta_\tau| \sqrt{\varepsilon}$ is what allows the argument to handle all signs of η_τ uniformly: when $\eta_\tau \ge 0$ the simpler estimate $\eta_\tau \|\boldsymbol{e}_r\|^2 / \phi_\varepsilon \le \eta_\tau \phi_\varepsilon$ already suffices, but when $\eta_\tau < 0$ this naive bound goes the wrong way and the $|\eta_\tau| \sqrt{\varepsilon}$ correction is needed. Combining gives,

$$\phi_\varepsilon'(r) \le \eta_\tau \phi_\varepsilon(r) + C_\tau + |\eta_\tau| \sqrt{\varepsilon}. \tag{57}$$

Inequality (57) is a linear differential inequality with constant coefficient η_τ and forcing $K_\varepsilon := C_\tau + |\eta_\tau|\sqrt{\varepsilon}$. We integrate by the standard integrating-factor argument (the linear instance of Grönwall; see e.g. 61, Lemma 5.16). Multiplying both sides of (57) by $e^{-\eta_\tau r} > 0$,

$$\frac{\mathrm{d}}{\mathrm{d}r}\big(e^{-\eta_\tau r}\phi_\varepsilon(r)\big) = e^{-\eta_\tau r}\big(\phi_\varepsilon'(r) - \eta_\tau\phi_\varepsilon(r)\big) \le K_\varepsilon e^{-\eta_\tau r}.$$

Integrating from 0 to r and using $\phi_\varepsilon(0) = \sqrt{\varepsilon}$,

$$e^{-\eta_\tau r}\phi_\varepsilon(r) - \sqrt{\varepsilon} \le K_\varepsilon \int_0^r e^{-\eta_\tau s}\,\mathrm{d}s.$$

A direct computation gives

$$\int_0^r e^{-\eta_\tau s}\,\mathrm{d}s = \begin{cases} \dfrac{1-e^{-\eta_\tau r}}{\eta_\tau}, & \eta_\tau \neq 0, \\ r, & \eta_\tau = 0, \end{cases} \quad = e^{-\eta_\tau r}\,\Psi_{\eta_\tau}(r),$$

where the second equality uses the definition of Ψ_{η_τ} in (48). Multiplying through by $e^{\eta_\tau r}$,

$$\phi_\varepsilon(r) \le \sqrt{\varepsilon}\,e^{\eta_\tau r} + K_\varepsilon\,\Psi_{\eta_\tau}(r), \qquad r \in [0, 1-\tau].$$

Since $\|\boldsymbol{e}_r\| \le \phi_\varepsilon(r)$ for every $\varepsilon > 0$, and since $K_\varepsilon = C_\tau + |\eta_\tau|\sqrt{\varepsilon} \to C_\tau$ as $\varepsilon \downarrow 0$, the right-hand side is continuous in ε at 0 for each fixed r, with

$$\lim_{\varepsilon\downarrow 0}\big(\sqrt{\varepsilon}\,e^{\eta_\tau r} + K_\varepsilon\Psi_{\eta_\tau}(r)\big) = C_\tau\Psi_{\eta_\tau}(r).$$

Therefore $\|\boldsymbol{e}_r\| \le C_\tau\Psi_{\eta_\tau}(r)$.

Substituting back $r = 1 - t$ proves (49). The supremum bound (50) is immediate from Lemma E.3 (i). □

Corollary E.6 (Stability under globally Lipschitz drifts)**.** *Suppose $\boldsymbol{v}$ and g are globally Lipschitz in $\boldsymbol{f}$, uniformly in $t \in [\tau, 1]$, with constants L_v and L_g respectively. Then Proposition E.5 holds with*

$$\eta_\tau = L_\tau := L_v + \tfrac{1}{2}B_\tau L_g \ge 0.$$

Consequently,

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le \frac{B_\tau\varepsilon_\tau}{2L_\tau}\big(e^{L_\tau(1-t)} - 1\big) \quad (L_\tau > 0),$$

whereas, if $L_\tau = 0$, then

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le \tfrac{1}{2}B_\tau\varepsilon_\tau(1-t).$$

Proof. Follows immediately from Remark E.4 (i) and Proposition E.5. □

Corollary E.7 (Stability under strongly monotone guidance)**.** *Assume the whitened setting $\boldsymbol{v} \equiv 0$, with $\beta(t) \ge \underline{B}_\tau > 0$ on $[\tau, 1]$, and suppose g satisfies*

$$\langle \boldsymbol{x} - \boldsymbol{y},\ g(t,\boldsymbol{x}) - g(t,\boldsymbol{y})\rangle \le -\mu_\tau\|\boldsymbol{x} - \boldsymbol{y}\|^2, \qquad \forall t \in [\tau, 1],\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m,$$

for some $\mu_\tau > 0$. Then Proposition E.5 applies with

$$\eta_\tau = -\tfrac{1}{2}\underline{B}_\tau\mu_\tau < 0,$$

and

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le \frac{B_\tau\varepsilon_\tau}{\underline{B}_\tau\mu_\tau}\left(1 - e^{-\frac{1}{2}\underline{B}_\tau\mu_\tau(1-t)}\right), \qquad t \in [\tau, 1].$$

In particular,

$$\sup_{t\in[\tau,1]}\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \le \frac{B_\tau\varepsilon_\tau}{\underline{B}_\tau\mu_\tau},$$

so the perturbation saturates at a level proportional to $\varepsilon_\tau/\mu_\tau$, for fixed truncated-interval constants.

Proof. With $\boldsymbol{v} \equiv 0$,

$$\boldsymbol{a}(r, \boldsymbol{f}) = \tfrac{1}{2}\beta(1-r)g(1-r, \boldsymbol{f}),$$

so for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ and $r \in [0, 1-\tau]$,

$$\langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{a}(r,\boldsymbol{x}) - \boldsymbol{a}(r,\boldsymbol{y})\rangle = \tfrac{1}{2}\beta(1-r)\langle \boldsymbol{x}-\boldsymbol{y}, g(1-r,\boldsymbol{x}) - g(1-r,\boldsymbol{y})\rangle.$$

By the strong monotonicity of g and $\beta(1-r) \geq \underline{B}_\tau > 0$,

$$\langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{a}(r,\boldsymbol{x}) - \boldsymbol{a}(r,\boldsymbol{y})\rangle \leq -\tfrac{1}{2}\underline{B}_\tau \mu_\tau \|\boldsymbol{x}-\boldsymbol{y}\|^2.$$

Hence (47) holds with $\eta_\tau = -\frac{1}{2}\underline{B}_\tau\mu_\tau < 0$. Substituting this value into Proposition E.5 and using $\Psi_\eta(r) = (1-e^{-|\eta|r})/|\eta|$ for $\eta < 0$ (Lemma E.3 (iii)) gives the required bound. □

Remark E.8 (Comparison of regimes). The unified bound (49) interpolates between three qualitatively distinct behaviours, governed by the sign of η_τ.

- *Expansive case* ($\eta_\tau > 0$). The bound grows exponentially in the backward integration horizon $1-t$. This is the worst-case behaviour obtained from a purely global Lipschitz argument.
- *Borderline case* ($\eta_\tau = 0$). The perturbation accumulates linearly:

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \leq C_\tau(1-t).$$

- *Contractive case* ($\eta_\tau < 0$). The pathwise error satisfies

$$\|\boldsymbol{f}_t - \widehat{\boldsymbol{f}}_t\| \leq \frac{C_\tau}{|\eta_\tau|}\left(1 - e^{-|\eta_\tau|(1-t)}\right) \leq \frac{C_\tau}{|\eta_\tau|}.$$

  Thus the approximation error saturates rather than accumulating indefinitely over the backward integration horizon. The constants may still depend on the truncation level τ.

The one-sided Lipschitz analysis can therefore yield sharper bounds than a global Lipschitz analysis whenever the effective reversed-time drift is monotone or contractive. In the GP setting, this behaviour is expected when the effective reversed-time drift is contractive over the region visited by the sampler. In whitened coordinates this can occur when the smoothed conditioning likelihood is uniformly strongly log-concave over that region.

E.2 Adding deterministic ODE discretisation error

In practice, the approximate ODE (45) is integrated numerically on a backward grid

$$1 = t_0 > t_1 > \cdots > t_N = \tau, \qquad h := \max_{0 \leq n \leq N-1}(t_n - t_{n+1}).$$

We assume that the chosen solver is a deterministic explicit Runge–Kutta-type method (e.g. explicit Euler, Heun's method, classical RK4, or one of the diffusion-tailored variants such as DPM-Solver [66], all of which are deterministic schemes for deterministic ODEs; see 10, Chapter 9 for a survey). The choice of solver enters through its global error along the relevant trajectory. Here $\widehat{\boldsymbol{f}}_t$ denotes the exact solution of the approximately guided ODE (45), whereas $\widehat{\boldsymbol{f}}^h_{t_n}$ denotes its numerical approximation on the grid. We assume the solver has global order q:

$$\max_{0 \leq n \leq N} \|\widehat{\boldsymbol{f}}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\| \leq C_{\mathrm{num},\tau}\, h^q. \tag{58}$$

Such bounds are classical in numerical analysis; for example, $q = 1$ for explicit Euler, $q = 2$ for Heun's method, $q = 4$ for RK4 [67, Section 2.6], and $q \geq 2$ for the DPM-Solver family [66].[3]

Proposition E.9 (Total error: guidance approximation plus discretisation)**.** *Under Assumption E.2, the existence assumptions of Proposition E.5, and the numerical global-error bound* (58)*, for every grid point $t_n \in [\tau, 1]$,*

$$\|\boldsymbol{f}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\| \leq C_\tau\, \Psi_{\eta_\tau}(1-t_n) + C_{\mathrm{num},\tau}\, h^q.$$

Consequently,

$$\max_{0 \leq n \leq N} \|\boldsymbol{f}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\| \leq C_\tau\, \Psi_{\eta_\tau}(1-\tau) + C_{\mathrm{num},\tau}\, h^q.$$

[3]The constant $C_{\mathrm{num},\tau}$ depends on the truncated interval $[\tau, 1]$, on bounds for $\widehat{\boldsymbol{a}}$ and its higher derivatives along the trajectory, and on the solver. We use q for the solver order to avoid clashing with the Wasserstein exponent p.

Proof. By the triangle inequality applied to the three flows $\boldsymbol{f}_{t_n} \to \widehat{\boldsymbol{f}}_{t_n} \to \widehat{\boldsymbol{f}}^h_{t_n}$,

$$\|\boldsymbol{f}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\| \leq \underbrace{\|\boldsymbol{f}_{t_n} - \widehat{\boldsymbol{f}}_{t_n}\|}_{\leq C_\tau \Psi_{\eta_\tau}(1-t_n) \text{ by Prop. E.5}} + \underbrace{\|\widehat{\boldsymbol{f}}_{t_n} - \widehat{\boldsymbol{f}}^h_{t_n}\|}_{\leq C_{\mathrm{num},\tau} h^q \text{ by (58)}}.$$

The supremum bound follows from monotonicity of Ψ_{η_τ} (Lemma E.3 (i)). □

Remark E.10 (Self-normalised Monte Carlo guidance). The Monte Carlo estimator (18) is a self-normalised importance sampler and is therefore biased at finite S. At a fixed deterministic state $(t, \boldsymbol{f})$, standard self-normalised importance sampling results give, under appropriate moment conditions on the weights and weighted scores,

$$\mathbb{E}[\widehat{g}_S(t, \boldsymbol{f})] - g(t, \boldsymbol{f}) = \mathcal{O}(S^{-1}), \qquad \mathrm{Var}(\widehat{g}_S(t, \boldsymbol{f})) = \mathcal{O}(S^{-1}),$$

with constants depending on the discrepancy between $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ and $p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{C}, \mathcal{D})$, for example through importance-weight moment or chi-squared-divergence quantities.

These pointwise Monte Carlo statements do not by themselves imply a trajectory-level $\mathcal{O}(S^{-1/2})$ error bound for the sampler. The numerical trajectory is evaluated at states that depend on the realised Monte Carlo guidance field. This dependence is particularly explicit when common random numbers are drawn once before integration and then reused across ODE steps. In that case, the trajectory is coupled to the empirical error field

$$(t, \boldsymbol{f}) \mapsto \widehat{g}_S(t, \boldsymbol{f}) - g(t, \boldsymbol{f}).$$

A stochastic version of Theorem E.1 would therefore require additional control of a uniform empirical-process quantity, such as

$$\mathbb{E}\left[\sup_{(t,\boldsymbol{f})\in\mathcal{R}_\tau} \|\widehat{g}_S(t, \boldsymbol{f}) - g(t, \boldsymbol{f})\|\right] \quad \text{or a corresponding high-probability bound.}$$

Such a bound need not follow from the pointwise variance rate without further assumptions on the likelihood, the importance weights, and the complexity of the class of guidance functions encountered along the trajectory. Accordingly, the perturbation bound above is used pathwise: conditional on a realised approximate guidance field satisfying Assumption E.2 (4), the resulting trajectory error is controlled by Theorem E.1. Establishing sharp stochastic trajectory-level Monte Carlo rates is left for future work.

F Practical algorithm details

F.1 Schedules

We define a VP diffusion process over whitened samples. The forward process marginals are

$$\hat{\mathbf{f}}_t = \alpha(t)\,\hat{\mathbf{f}}_0 + \sqrt{1-\alpha^2(t)}\,\boldsymbol{z}, \qquad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m), \tag{59}$$

where the signal-to-noise decay is governed by a linear β-schedule,

$$\beta(t) = \beta_0 + (\beta_1 - \beta_0)\,t, \qquad \alpha(t) = \exp\!\left(-\frac{1}{2}\int_0^t \beta(s)\,\mathrm{d}s\right) = \exp\!\left(-\tfrac{1}{2}\beta_0 t - \tfrac{1}{4}(\beta_1-\beta_0)t^2\right), \tag{60}$$

with $\beta_0 = 10^{-5}$ and $\beta_1 = 10.0$. These endpoints are chosen to ensure $\alpha(1) \approx 0$ so that the initial condition is effectively pure noise, whilst ensuring that $\alpha(0) = 1$ so that the terminal distribution recovers the GP posterior exactly.

F.2 Time Discretisation

We integrate from $t = 1$ to $t \approx 0$ using a schedule that is uniform in *log-SNR* space, where $\mathrm{SNR}(t) = \alpha(t)/\sqrt{(1-\alpha^2(t) + 1\times 10^{-8})}$. Binary search is used to invert the SNR function and place the $T+1$ grid points $t_0 = 1 > t_1 > \cdots > t_T \approx 0$ at equal intervals in $\log \mathrm{SNR}$. This concentrates steps where the SNR changes most rapidly, improving integration quality without increasing T.

F.3 Practical Guidance Calculation

To reduce variance between ODE steps, we use the reparameterisation trick, sampling S isotropic Gaussian samples $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ before beginning integration. At each ODE step, we

1. Build S samples $\boldsymbol{f}_0^{(i)} \sim p(\boldsymbol{f}_0 \mid \boldsymbol{f}_t, \mathcal{D})$ via $\boldsymbol{f}_0^{(i)} = \boldsymbol{\mu}_{0|t} + \boldsymbol{\Sigma}_{0|t}^{1/2}\boldsymbol{\epsilon}^{(i)}$, with $\boldsymbol{\mu}_{0|t}$ from (22) and $\boldsymbol{\Sigma}_{0|t}$ from (23) with $\mathbf{m}_{*|\boldsymbol{y}}$, $\mathbf{K}_{**|\boldsymbol{y}}$ and $\mathbf{A}_{|\boldsymbol{y}}(t)$ in place of $\mathbf{m}_*, \mathbf{K}_{**}$ and $\mathbf{A}(t)$.
2. Evaluate each log-likelihood $\log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$ and its gradient $\nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)})$ at each sample,
3. Compute normalised weights via the numerically stable log-sum-exp operation $\log \bar{w}^{(i)} = \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}) - \operatorname{logsumexp}_r \log p(\mathcal{C} \mid \boldsymbol{f}_0^{(r)})$. Here
$$\operatorname{logsumexp}_r a^{(r)} := \max_r a^{(r)} + \log\left[\sum_r \exp\left(a^{(r)} - \max_r a^{(r)}\right)\right]$$
avoids overflow and underflow by subtracting the potentially very small maximal value.
4. Evaluate the weighted sum (30), applying the Jacobian to each gradient term,
5. We clip the norm of the vector field (after scaling by $-\frac{1}{2}\beta(t)$ as prescribed by the probability flow ODE (16)) to limit excessively large steps to ensure stable integration. We use the smooth saturation: $\boldsymbol{v} \mapsto \boldsymbol{v} \cdot \tau \tanh(\|\boldsymbol{v}\|/\tau)/(\|\boldsymbol{v}\| + 1e^{-8})$, where $\tau = 1 \times 10^2$ is a maximum norm threshold. This transformation bounds excessively large gradients whilst preserving Lipschitz continuity.

F.4 Full integration loop

The full sampling loop integrates the probability-flow ODE with the Euler method. Since the unconditional vector field is zero in whitened coordinates, each step reduces to applying only the conditional correction, see Algorithm 1.

G Experimental details for monotonic and bounded Bayesian regression

Task. We demonstrate shape-constrained GP regression, conditioning a 1D GP on noisy observations while simultaneously enforcing that the inferred function is (i) *monotonically increasing* and (ii) *bounded* between a known lower and upper envelope. This setting is representative of problems in dose-response modelling, reliability analysis, or any domain where monotonicity and range constraints are required.

Data. The ground truth function is
$$f_{\text{true}}(x) = \frac{1}{3}[\arctan(20x - 10) - \arctan(-10)], \quad x \in [0, 1]. \tag{61}$$
We observe $n = 7$ noise-free evaluations at locations $x_i = 0.1 + 1/(i+1)$ for $i = 1, \ldots, 7$, generated according to (1) with $\boldsymbol{\Gamma} = \sigma^2 \mathbf{I}_n$ and measurement-error variance $\sigma^2 = 10^{-10}$ (i.e. effectively noiseless observations).

Shape constraints. Our target is $f_0(\cdot)$ evaluated on a uniform grid of size $m = 64$ on $[0, 1]$; we denote the collection of elements by $\boldsymbol{f}_0$. We make two constraints $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$ when making inference on $\boldsymbol{f}_0$:

Monotonicity. We require $f_0'(x) \geq 0$ everywhere on $[0, 1]$. This is implemented on our uniform grid using constraints on the forward finite differences
$$c_i = \frac{f_{0,i+1} - f_{0,i}}{\Delta x}, \quad i = 1, \ldots, m - 1, \tag{62}$$
where $\Delta x = 1/m$. We implement these constraints through a probit log-probability model (via a standard-normal CDF relaxation):
$$\log p(\mathcal{C}_1 \mid \mathbf{f}_0) = \sum_{i=1}^{m-1} \log \Phi\left(\frac{c_i}{v}\right), \tag{63}$$

Algorithm 1 FLOWGP: sampling from a GP predictive distribution under arbitrary conditioning via the whitened probability-flow ODE

Require: Linear-Gaussian predictive mean $\mathbf{m}_{*|\boldsymbol{y}}$ and covariance $\mathbf{K}_{**|\boldsymbol{y}}$ from the data $\mathcal{D}$; non-Gaussian condition $\mathcal{C}$ with point-wise evaluable likelihood $p(\mathcal{C} \mid \boldsymbol{f}_0)$; number of ODE steps T; number of Monte Carlo samples S; smooth-clipping threshold $v_{\max}$; schedule $\alpha(t), \beta(t)$ (Appendix F.1).

Ensure: Sample $\boldsymbol{f}_0 \sim p(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C})$.

Factorise $\mathbf{K}_{**|\boldsymbol{y}} = \mathbf{L}\mathbf{L}^\intercal$ {one-off $\mathcal{O}(m^3)$ cost; defines $\mathcal{W}^{-1}(\cdot) = \mathbf{L} \cdot + \mathbf{m}_{*|\boldsymbol{y}}$}
Build decreasing time grid $1 = t_0 > t_1 > \cdots > t_T \approx 0$, uniform in log-SNR (Appendix F.2)
Draw initial whitened state $\hat{\boldsymbol{f}}_{t_0} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ {$t = 1$: pure white noise}

for $j = 0, 1, \ldots, T-1$ **do**
  $\Delta t_j \leftarrow t_j - t_{j+1}, \quad \alpha_j \leftarrow \alpha(t_j), \quad \beta_j \leftarrow \beta(t_j)$

  {(i) Draw S samples from $\hat{\boldsymbol{f}}_0 \mid \hat{\boldsymbol{f}}_{t_j}, \mathcal{D} \sim \mathcal{N}\big(\alpha_j\, \hat{\boldsymbol{f}}_{t_j},\, (1-\alpha_j^2)\mathbf{I}_m\big)$; see (25).}
  for $i = 1, \ldots, S$ **do**
    $\hat{\boldsymbol{f}}_0^{(i)} \sim \mathcal{N}\big(\alpha_j\, \hat{\boldsymbol{f}}_{t_j},\, (1-\alpha_j^2)\mathbf{I}_m\big)$
    $\boldsymbol{f}_0^{(i)} \leftarrow \mathbf{L}\, \hat{\boldsymbol{f}}_0^{(i)} + \mathbf{m}_{*|\boldsymbol{y}}$ {unwhiten before evaluating the likelihood}
  end for

  {(ii) Self-normalised importance weights (numerically stable form)}
  $\ell^{(i)} \leftarrow \log p\big(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}\big)$ for $i = 1, \ldots, S$
  $\bar{w}^{(i)} \leftarrow \exp\Big(\ell^{(i)} - \operatorname{logsumexp}_r \ell^{(r)}\Big)$ for $i = 1, \ldots, S$

  {(iii) Likelihood scores via automatic differentiation through $\boldsymbol{f}_0^{(i)} = \mathbf{L}\hat{\boldsymbol{f}}_0^{(i)} + \mathbf{m}_{*|\boldsymbol{y}}$}
  $\boldsymbol{s}^{(i)} \leftarrow \nabla_{\hat{\boldsymbol{f}}_0^{(i)}} \log p\big(\mathcal{C} \mid \boldsymbol{f}_0^{(i)}\big)$ for $i = 1, \ldots, S$

  {(iv) Guided velocity field in whitened space (Eq. (19) with MC guidance from Eq. (18))}
  $\boldsymbol{v} \leftarrow -\frac{1}{2}\, \beta_j\, \alpha_j \sum_{i=1}^{S} \bar{w}^{(i)}\, \boldsymbol{s}^{(i)}$

  {(v) Smooth clipping of the step to promote numerical stability}
  $\boldsymbol{v} \leftarrow \dfrac{v_{\max}\, \tanh\big(\|\boldsymbol{v}\| / v_{\max}\big)}{\|\boldsymbol{v}\|}\, \boldsymbol{v}$

  {(vi) Explicit Euler step of the probability-flow ODE, integrating from $t = 1$ to $t = 0$}
  $\hat{\boldsymbol{f}}_{t_{j+1}} \leftarrow \hat{\boldsymbol{f}}_{t_j} - \Delta t_j\, \boldsymbol{v}$

end for
return $\boldsymbol{f}_0 \leftarrow \mathbf{L}\, \hat{\boldsymbol{f}}_{t_T} + \mathbf{m}_{*|\boldsymbol{y}}$ {unwhiten final state}

where $\mathcal{C}_1 = \{c_1, \ldots, c_{m-1}\}$, $\Phi(\cdot)$ is the standard normal CDF, and the sharpness parameter $v = 10^{-4}$. The model heavily penalises for any c_i being negative.

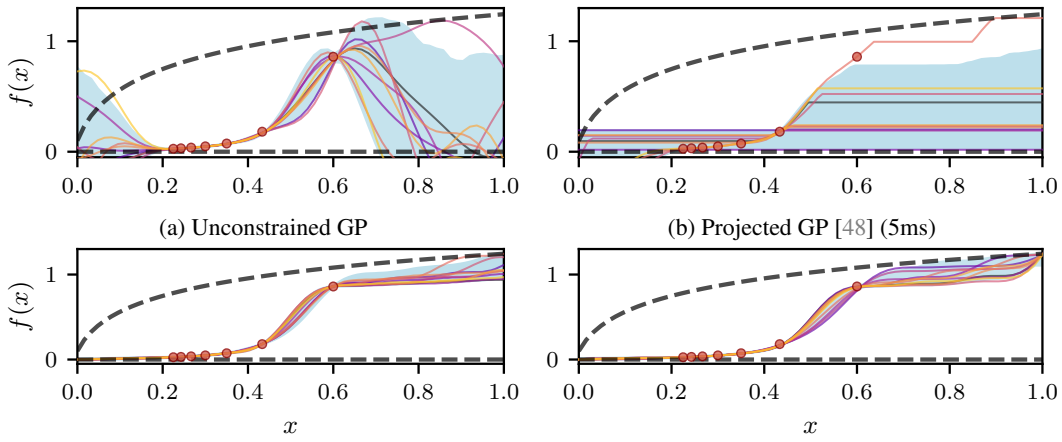
Boundedness. We require $\ell(x) \le f_0(x) \le u(x)$ for all x, where the lower and upper envelopes are

$$\ell(x) = 0, \qquad u(x) = \frac{1}{3}\log(30x + 1) + 0.1. \tag{64}$$

We implement this constraint on $\boldsymbol{f}_0$ by considering the $2m$ margin values $\mathcal{C}_2 = \{(u(x_i) - f_{0,i},\, f_{0,i} - \ell(x_i)) : i = 1, \ldots, m\}$. These are modelled using the probit relaxation of (63) with $v = 10^{-5}$ and $\mathcal{C}_1$ replaced with $\mathcal{C}_2$.

GP prior. We use a GP prior with a squared exponential kernel over functions on $[0, 1]$ of the form

$$\operatorname{cov}(f_0(x), f_0(x')) = \tau^2 \exp\left(\frac{-(x - x')^2}{2\kappa^2}\right)$$



(a) Unconstrained GP

(b) Projected GP [48] (5ms)

(c) Posterior Projected GP [49] (40ms)

(d) FLOWGP (Ours) (200ms)

Figure 4: Predictive samples from an unconstrained GP **(a)**, two custom projection methods **(b)** and **(c)** [48, 49], and FLOWGP **(d)**, for a bounded monotonic regression problem. All methods are conditioned on seven observations (red dots) and we wish to encode monotonicity and bound constraints (dashed lines). FLOWGP remains prior-consistent and matches the quality of the bespoke method, without a substantial increase in runtime. We plot 10 predictive samples and the 0.05-0.95 quantiles from 100 samples. Timings are all on an Apple M2 Pro processor with 32GB RAM.

, with fixed length-scale $\kappa = 0.1$ and scaling parameter $\tau^2 = 0.25$ (assumed known). The GP evaluated on our grid is conditioned on the 7 observations to obtain the posterior mean $\mathbf{m}_{*|\boldsymbol{y}}$ and covariance $\mathbf{K}_{**|\boldsymbol{y}}$ on the $m = 64$ grid points, yielding the base Gaussian predictive distribution $\mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$.

Additional results. Figure 4 shows the predictions from FLOWGP along with those from an unconstrained GP and two projection methods [48, 49]. FLOWGP closely matches the quality of [49], a bespoke method specifically engineered for this constraint structure whilst [48] exhibits prior inconsistency.

H Experimental details for physics-informed experiments

H.1 Evaluation strategy

Evaluating sample paths at new test points. FLOWGP produces sample paths evaluated on a fixed discretisation grid, whereas the test observations lie at arbitrary locations $\mathbf{x}^*$. Therefore, we extend our sampled field across new test points by kernel smoothing. In particular, we can apply the standard GP regression formula to get a smoothed field at $\mathbf{x}^{new}$. Of course, we can longer guarantee that our condition will hold at $\mathbf{x}^{new}$, but we can ensure that it is at least consistent with the closed-form GP components. In particular, we extend a sample $\boldsymbol{f}^{(s)}$ defined over a grid $\boldsymbol{X}_*$ to a new location via

$$\hat{\boldsymbol{f}}^{(s)}(\mathbf{x}^{new}) \;=\; \mu_{*|y}(\boldsymbol{x}^{\text{new}}) \;+\; \mathbf{k}(\mathbf{x}^{new}, \boldsymbol{X}_*)\, \boldsymbol{K}_{**|y}^{-1}\big(\boldsymbol{f}^{(s)} - \boldsymbol{m}_{*|y}\big), \tag{65}$$

where $\mu_{*|y}(\cdot) \in \mathbb{R}^1$ is the GP's posterior mean at $\mathbf{x}^{new}$ and $\mathbf{k}_{*|y}(\mathbf{x}^*, \boldsymbol{X}_*) \in \mathbb{R}^{1\times m}$ is the vector of posterior cross-covariances between at $\mathbf{x}^{new}$ and all the grid points $\boldsymbol{X}_*$, and $\mathbf{K}_{**|y}$ and $\boldsymbol{m}_{*|y}$ are the covariance and mean of $\boldsymbol{f}_0|\mathcal{D}$.

Metrics. From the ensemble we form a Monte Carlo predictive distribution. The predictive mean and variance at test point i are

$$\hat{\mu}_i = \frac{1}{S}\sum_{s=1}^{S} \hat{f}^{(s)}(\mathbf{x}_i^*), \qquad \hat{\sigma}_i^2 = \frac{1}{S-1}\sum_{s=1}^{S}\big(\hat{f}^{(s)}(\mathbf{x}_i^*) - \hat{\mu}_i\big)^2 + \sigma_\varepsilon^2, \tag{66}$$

where σ^2 is the observation noise variance (if applicable). We report two scalar metrics averaged over the M test points:

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{\mu}_i - y_i)^2}, \tag{67}$$

$$\mathrm{NLPD} = \frac{1}{M}\sum_{i=1}^{M}\left[\tfrac{1}{2}\log(2\pi\hat{\sigma}_i^2) + \frac{(y_i - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2}\right], \tag{68}$$

where y_i is the observed target. RMSE measures point-prediction accuracy, while NLPD (negative log predictive density) also penalises miscalibrated uncertainty. Uncertainty in both metrics is reported as ± 1 standard deviation estimated by bootstrapping over the 1000 samples.

H.2 Damped Pendulum Experiment

Task. We consider the problem of inferring the angular displacement trajectory $\theta(t)$ of a damped nonlinear pendulum from noisy sparse observations, while simultaneously enforcing that the inferred trajectory satisfies the governing equation of motion. The pendulum dynamics are described by the second-order ODE

$$\theta''(t) + \sin\big(\theta(t)\big) + \beta\,\theta'(t) = 0, \tag{69}$$

where $\beta = 0.2$ is the damping coefficient. The time horizon is $t \in [0, 30]$ s (note that here and below we abuse notation somewhat and let 't' also index our input space).

Data. We use a fixed train/test split generated from a numerical solution of Equation (69). Training observations are noisy evaluations of $\theta(\cdot)$ at irregularly spaced time points. The test set is evaluated on a subsampled grid of held-out time points. Both training and test data are as provided in the PHYSS [36].

Experimental settings. We run two settings with different purposes:

- **Qualitative / plotting setting.** Measurement-error variance $\sigma^2 = 0.15^2$. The GP posterior and samples are evaluated on a grid of $m = 250$ points. We generate 25 predictive samples. This setting is used to produce visualisations.
- **Quantitative / metrics setting.** Measurement-error variance $\sigma^2 = 0.01^2$. The grid is reduced to $m = 125$ points, and 1000 predictive samples are drawn to obtain reliable estimates of predictive means and variances. This setting is used to compute RMSE and NLPD.

GP prior. We place a GP prior over trajectories $f_0 : [0, 1] \to \mathbb{R}$ (time is normalised by dividing by 30) using a scaled RBF kernel

$$\mathrm{cov}(f_0(t), f_0(t')) = \tau^2 \exp\left(-\frac{(t - t')^2}{2\kappa^2}\right), \tag{70}$$

with hyperparameters τ^2 and κ optimised together with an affine mean function by maximising the marginal likelihood using just $\mathcal{D}$,. As in the previous experiment, the GP predictive mean $\mathbf{m}_{*|\boldsymbol{y}}$ and covariance $\mathbf{K}_{**|\boldsymbol{y}}$ on the evaluation grid are used to construct the base Gaussian predictive distribution $\mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$.

Physics-informed conditioning. The ODE constraint (69) is encoded as a Gaussian likelihood over the pointwise residuals of our target function $f_0(\cdot)$ on our prediction grid $t_1, \ldots, t_m$:

$$r(t_j) = f_0{}''(t_j) + \sin(f_0(t_j)) + \beta\, f_0{}'(t_j), \quad j = 2, \ldots, m - 1, \tag{71}$$

with a likelihood standard deviation parameter $\sigma_{\text{phys}} = 10^{-10}$. Note that, while the likelihood function is Gaussian, the constraints it encodes are nonlinear. We evaluate the derivatives using central finite differences (see Appendix H.4 for an example). Guidance is applied during probability-flow ODE denoising (1000 steps, 5 MC samples per step, whitened representation).

Evaluation. We form the predictive mean and variance from predictive samples, adding measurement error with variance σ^2 to the sample variance in order to get the predictive variance of test data. We report results using RMSE and NLPD.

H.3 Allen–Cahn Equation Experiment

Task. We consider the problem of inferring the solution field $u(x,t)$ of the Allen–Cahn equation from noisy, partial observations in an early-time window, while enforcing the PDE and symmetry boundary conditions over the full domain. The governing equation is

$$\frac{\partial u(x,t)}{\partial t} = \varepsilon \frac{\partial^2 u(x,t)}{\partial x^2} + 5u(x,t) - 5u(x,t)^3, \tag{72}$$

with $\varepsilon = 10^{-5}$, $x \in [-1,1]$, $t \in [0,1]$.

Data. Training observations are drawn from the early-time regime $t < 0.28$ (256 randomly subsampled points) and the test set contains 1000 randomly subsampled points from $t > 0.28$. Both training and test data splits are as provided in the PHYSS [36].

Experimental settings. We run two settings with different purposes:

- **Qualitative / plotting setting.** Measurement-error variance $\sigma^2 = (0.01)^2$. The field is discretised on a $H \times W = 100 \times 100$ grid ($m = 10{,}000$). We draw 5 predictive samples. This setting is used to produce visualisations.
- **Quantitative / metrics setting.** Effectively noiseless observations ($\sigma^2 = 10^{-10}$). The grid is reduced to $H \times W = 50 \times 20$ ($m = 1000$). We draw 10 predictive samples. This setting is used to compute RMSE and NLPD.

GP prior. We use a squared-exponential covariance function 2D field $f_0 : [-1,1] \times [0,1] \to \mathbb{R}$

$$\operatorname{cov}(f_0(x,t), f_0(x',t')) = \tau^2 \exp\left(-\frac{(x-x')^2}{2\kappa_x^2} - \frac{(t-t')^2}{2\kappa_t^2} \right), \tag{73}$$

with the length scales and variances fitted by maximising the marginal likelihood using just $\mathcal{D}$. As in the previous experiments, the GP predictive mean $\mathbf{m}_{*|\boldsymbol{y}}$ and covariance $\mathbf{K}_{**|\boldsymbol{y}}$ on the evaluation grid are used to construct the base Gaussian predictive distribution $\mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$.

Physics-informed conditioning. Samples are conditioned jointly on the Allen–Cahn PDE residual (evaluated at interior spatial rows, via central finite differences in both x and t) and on symmetric boundary conditions enforcing $f_0(-1,t) = f_0(1,t)$ and matching first spatial derivatives at the boundaries (see Appendix H.4 for an example). Both conditions use $\sigma_{\text{phys}} = 10^{-5}$, and are combined as a product when constructing $p(\mathcal{C} \mid \boldsymbol{f}_0)$.

Evaluation. For each test point (x_j, t_j) we form the predictive mean and variance from the predictive samples, before adding observation noise σ^2 to the predictive variance. We report our results using RMSE and NLPD.

H.4 Viscous Burgers' Equation Experiment

Task. We consider the problem of inferring the velocity field $u(x,t)$ of the one-dimensional viscous Burgers' equation from a small number of observations of the initial condition, while enforcing that the inferred field satisfies the PDE and homogeneous Dirichlet boundary conditions over the full spatio-temporal domain (as considered in [38]). The governing equation is

$$\frac{\partial u(x,t)}{\partial t} + u \frac{\partial u(x,t)}{\partial x} = \nu \frac{\partial^2 u(x,t)}{\partial x^2}, \tag{74}$$

where $\nu = 0.02$ is the kinematic viscosity. The spatial domain is $x \in [-1,1]$ and the time domain is $t \in [0,1]$.

Data and experimental variants. We assume we have noisy observations of the initial condition that are generated from the true profile

$$u(x, 0) = -\sin\big(\pi(2x - 1)\big), \tag{75}$$

at uniformly spaced interior points. We consider two experimental settings:

- **Noisy, scarce data.** $n = 5$ observations are used, each corrupted by independent Gaussian measurement error with variance $\sigma^2 = (0.01)^2$. This setting tests the ability to recover the PDE solution from very few, noisy measurements.
- **Noiseless, dense data.** $n = 100$ observations are used, with effectively zero measurement error ($\sigma^2 = 10^{-12}$). This setting tests physics enforcement when the initial condition is almost exactly known on a dense grid.

In both cases the observations are 2D input pairs $(x_j, 0)$ covering only the initial time slice; the model must extrapolate the dynamics forward in time by satisfying the PDE constraint. Ground-truth solutions at three evaluation snapshots $t \in \{0.2, 0.5, 0.8\}$ are used for quantitative comparison.

GP prior. We place a Gaussian process prior over the 2D field $f_0 : [-1, 1] \times [0, 1] \to \mathbb{R}$. We use a squared exponential kernel

$$\operatorname{cov}(f_0(x, t), f_0(x', t')) = \tau^2 \exp\left(-\frac{(x - x')^2}{2\kappa_x^2} - \frac{(t - t')^2}{2\kappa_t^2}\right), \tag{76}$$

with fixed length-scales $\kappa_x = 0.025$, $\kappa_t = 0.3$, and output scale $\tau^2 = 1.0$ (assumed known, following the benchmark protocol of [38]).

We construct the GP predictive mean $\mathbf{m}_{*|\boldsymbol{y}}$ and covariance $\mathbf{K}_{**|\boldsymbol{y}}$ on a uniform $H \times W = 50 \times 20$ grid, giving a base Gaussian predictive distribution $\mathcal{N}(\mathbf{m}_{*|\boldsymbol{y}}, \mathbf{K}_{**|\boldsymbol{y}})$ with $m = HW = 1000$.

Physics-informed conditioning. We condition samples jointly on the Burgers' PDE residual and on homogeneous Dirichlet boundary conditions.

PDE residual. Given a candidate field $\mathbf{f}_0 \in \mathbb{R}^{H \times W}$ with spacings $\Delta x = 2/(H - 1)$ and $\Delta t = 1/(W - 1)$, we compute derivatives via central finite differences at interior points:

$$\left.\frac{\partial f_0}{\partial t}\right|_{i,j} \approx \frac{f_{0,i,j+1} - f_{0,i,j-1}}{2\Delta t}, \qquad j = 1, \ldots, W - 2, \tag{77}$$

$$\left.\frac{\partial f_0}{\partial x}\right|_{i,j} \approx \frac{f_{0,i+1,j} - f_{0,i-1,j}}{2\Delta x}, \qquad i = 1, \ldots, H - 2, \tag{78}$$

$$\left.\frac{\partial^2 f_0}{\partial x^2}\right|_{i,j} \approx \frac{f_{0,i+1,j} - 2f_{0,i,j} + f_{0,i-1,j}}{\Delta x^2}. \tag{79}$$

The pointwise PDE residual is

$$r(x_j, t_j) = \left.\frac{\partial f_0}{\partial t}\right|_{i,j} + f_{0,i,j} \left.\frac{\partial f_0}{\partial x}\right|_{i,j} - \nu \left.\frac{\partial^2 f_0}{\partial x^2}\right|_{i,j}. \tag{80}$$

Boundary conditions. Homogeneous Dirichlet conditions $f_0(-1, t) = f_0(1, t) = 0$ for all t are enforced by pinning the top ($i = 0$) and bottom ($i = H - 1$) spatial rows to zero.

Both conditions are modelled as Gaussian likelihoods with $\sigma_{\text{phys}} = 10^{-5}$ (PDE) and $\sigma_{\text{bc}} = 10^{-6}$ (boundary), combined as a product when constructing $p(\mathcal{C} \mid \boldsymbol{f}_0)$. Unlike all other experiments in this paper, we found that FLOWGP required $10,000$ ODE steps (as opposed to $1,000$) to resolve the resulting flow.

Evaluation. We evaluate the conditioned samples at $t \in \{0.2, 0.5, 0.8\}$, reporting RMSE and NLPD at each snapshot (pooling all spatial points across all three temporal snapshots) for both experimental variants. Results shown in Table 3, where we follow the same formatting as in Table 1.

Model	RMSE	NLPD	Time	Model	RMSE	NLPD	Time
Kernel	**0.01**	N/A	$\mathbf{4{\cdot}10^1}$	Kernel	0.25	N/A	$\mathbf{4{\cdot}10^1}$
FLOW	0.04 (0.00)	**-1.82 (0.04)**	$8{\cdot}10^1$	FLOW	0.30 (0.01)	0.81 (0.05)	$8{\cdot}10^1$
FLOW_W	0.03 (0.00)	-1.35 (0.12)	$8{\cdot}10^1$	FLOW_W	**0.22** (0.00)	**-1.44 (0.03)**	$8{\cdot}10^1$

Table 3: The whitened formulation of FLOWGP matches existing physics-obeying deterministic kernel regression methods [38] in accuracy on the Burgers benchmarks with the additional benefit of providing confidence intervals. **left:** dense low-noise regime. **right:** sparse high-noise regime where the kernel approach degrades. We used the authors' open-source implementation, timing both theirs and our methods on our Nvidia A4500 workstation.

I Bayesian Optimisation Experiments

Problem setting We consider the BOPE setup of [54], and provide a short description below. For more details, we refer readers to the original paper, given that we consider exactly the same setting (with only a few minor differences introduced by FLOWGP, which we discuss in the next paragraph). In [54], an expensive and black-box function $\boldsymbol{f}_{\text{true}} : \mathbb{R}^d \to \mathbb{R}^k$ maps a decision variable $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ to k objective values, and a decision maker (DM) has an unknown utility $g_{\text{true}} : \mathbb{R}^k \to \mathbb{R}$ over these outputs. The goal is to find

$$\mathbf{x}^* \in \arg\max_{\mathbf{x}\in\mathcal{X}} g_{\text{true}}(\boldsymbol{f}_{\text{true}}(\mathbf{x})).$$

The procedure alternates between two stages. In *experimentation*, the experimenter selects $\mathbf{x}_t$ and observes $\boldsymbol{f}_{\text{true}}(\mathbf{x}_t)$, yielding a dataset $\mathcal{D} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$. In *preference exploration*, the DM is shown a pair of previously observed outputs $(\mathbf{y}_{1,t}, \mathbf{y}_{2,t})$ and returns $p_t \in \{-1, +1\}$. Restricting comparisons to observed outputs, rather than allowing arbitrary or unattainable pairs as in [68], avoids presenting the DM with options that cannot be realised. We model DM noise as utility noise:

$$p_t = 2 \cdot \mathbb{1}_{g_{\text{true}}(\mathbf{y}_{1,t}) - g_{\text{true}}(\mathbf{y}_{2,t}) + \epsilon > 0} - 1, \quad \epsilon \sim \mathcal{N}(0, \sigma^2_{\text{noise}}),$$

so error probability scales with the utility gap rather than being constant. After m comparisons, the preference data is $\mathcal{P}_t := \{(\mathbf{y}_{1,i}, \mathbf{y}_{2,i})\}_{i=1}^m$ with responses $\mathcal{R}_t := \{p_i\}_{i=1}^m$. The two stages alternate until a stopping condition is met, and the setting supports batch acquisition of both $\mathbf{x}_t$ and comparison pairs.

Experimental details For the experiments in Figure 5, we rely on the same set-up and codebase[4] as [54], modifying only the parts necessary for our FLOWGP model to work. Specifically, we rely on the same acquisition function, namely

$$\text{qNEIUU}(\mathbf{x}_{1:q}) = \frac{1}{M}\frac{1}{N}\sum_{j=1}^{M}\sum_{k=1}^{N}\left\{\max g^j\left(\boldsymbol{f}^k(\mathbf{x}_{1:q})\right) - \max g^j\left(\boldsymbol{f}^k(\mathbf{X}_n)\right)\right\}^+,$$

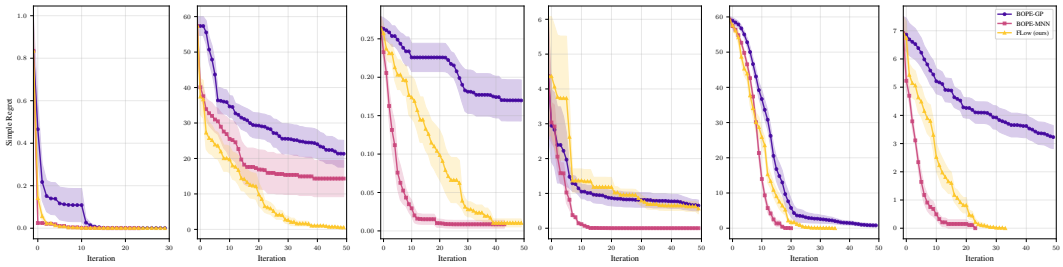
where $\boldsymbol{f}$ is the same GP surrogate as in [54]. For g, we replace their custom MoNNE architecture by our FLOWGP, where the underlying GP is the `PairwiseGP` model from `BoTorch` [69, 55] and $\mathcal{C}$ encodes monotonicity with respect to all dimensions of $\mathbf{x}$. After building the two surrogate models $\boldsymbol{f}$ and g, qNEIUU is computed by Monte Carlo simulation, where $\mathbf{x}_{1:q} \in \mathcal{X}^q$ (with $q = 1$ in this paper), $\mathbf{X}_n$ is the tensor of evaluated solutions, $\{\cdot\}^+$ denotes the positive part, M is the number of samples from g, N is the number of predictive samples from $\boldsymbol{f}$, $g^j(\cdot)$ is the j-th sample from g, and $\boldsymbol{f}^k(\cdot)$ is the k-th predictive sample from $\boldsymbol{f}$.

For the preference-exploration stage, we do not rely on the more complicated IEUBO presented in [54], and instead rely simply on EUBO [68]:

$$\text{EUBO}(\mathbf{y}_{1,i}, \mathbf{y}_{2,i}) = \frac{1}{M}\sum_{j=1}^{M}\max\left\{g^j(\mathbf{y}_{1,i}), g^j(\mathbf{y}_{2,i})\right\}, \quad \mathbf{y}_{1,i}, \mathbf{y}_{2,i} \in \{\mathbf{y}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_n\}.$$

Since the FLOWGP model does not easily allow for gradient-based optimisation of the acquisition function, we instead evaluate the acquisition function on a set of candidate points, from which we

[4] `https://github.com/HanyangHenry-Wang/BOPE-MoNNE`



(a) (b) (c) (d) (e) (f)

Figure 5: On all six Bayesian Optimisation with Preference Exploration (BOPE) problems considered in [54], our FLOWGP model improves upon the underlying preference-learning GP [55] by easily incorporating monotonicity, and often matches the problem-specific and carefully-engineered Monotonic Neural Network Ensemble (MoNNE) from [54]. **(a)** DTLZ2 (3D) + Linear (2D). **(b)** OSY (6D) + Quadratic (8D). **(c)** VehicleSafety (5D) + KumaraswamyCDF (3D). **(d)** VLMOP3 (2D) + Exponential (3D). **(e)** ZDT1 (10D) + LinearExponential (2D). **(f)** CarCabDesign (7D) + CobbDouglas (9D).

select the best one. To generate this set of candidate points, we rely on a technique similar to the state-of-the-art "Vanilla BO" method [70], which samples points both globally across the search space $\mathcal{X}$ and locally (as a Gaussian) around the incumbent.

J LLM-Guided Process Experiment

Task. Our final experiment demonstrates a *product-of-experts* construction that combines a GP stochastic-process prior with marginal predictive densities derived from a large language model (LLM). The LLM is queried with a natural-language description of the time series and returns a discrete probability distribution over output bins at each input location; this constitutes a text-conditioned "expert" density.

Product-of-experts formulation. Let $p(\boldsymbol{f}_0 \mid \mathcal{D})$ denote the GP predictive density and $q(\boldsymbol{f}_0 \mid \mathcal{C})$ the LLM-derived density conditioned on text prompt $\mathcal{C}$. We define the target distribution as the normalised product of experts

$$\pi(\boldsymbol{f}_0 \mid \mathcal{D}, \mathcal{C}) \propto p(\boldsymbol{f}_0 \mid \mathcal{D})\, q(\boldsymbol{f}_0 \mid \mathcal{C}). \tag{81}$$

Here, we are following exactly the setup of [57] and use the LLM-derived density provided by their code. Note that this product-of-experts formation is an alternative way to include additional knowledge into a GP, distinct from the likelihood conditioning used for all our other experiments.

We refer to their paper for details about the specific prompting strategy, number of tokens, temperature settings, and any post-processing of the Qwen3.5 [56] LLM's output.

Conditioning implementation. The LLM scores $q(\boldsymbol{f}_0 \mid \mathcal{C})$ are pre-computed offline and stored as per-timestep histograms over discrete output bins. In order to provide a smooth likelihood for our framework, we evaluate the log-likelihood of the current sample under a kernel-smoothed version of the LLM histogram density and add its gradient to the unconditional score.

Concretely, let $\boldsymbol{f}_0 \in \mathbb{R}^m$ be the current sample. For each grid point j and bin k with edges $[b_k^-, b_k^+]$ and LLM probability mass $p_k^{(j)}$, we compute

$$\log q(\boldsymbol{f}_0 \mid \mathcal{C}) = \sum_{j=1}^{m} \frac{1}{b_j^+ - b_j^-} \log \sum_k p_k^{(j)} \left[\Phi\left(\frac{b_k^+ - f_{0,j}}{\nu} \right) - \Phi\left(\frac{b_k^- - f_{0,j}}{\nu} \right) \right], \tag{82}$$

where $\Phi(\cdot)$ is the standard normal CDF and $\nu = 0.5$ is a smoothing bandwidth parameter. This replaces the discrete histogram with a continuous, differentiable density by treating each bin's mass as a Gaussian mixture component. Gradients of this expression with respect to $\boldsymbol{f}_0$ are computed analytically via the Gaussian PDF, so no backpropagation through the LLM is required at any point.

We build bins as all the numbers that are 3 significant figures and within the problem range (see below). A `BoundedCondition` is combined multiplicatively to keep samples within physically plausible output ranges.

Datasets and text prompts. We evaluate on three qualitative scenarios, each defined by a natural-language prompt, the Qwen3.5 [56] LLM, and a corresponding GP prior:

- **Bankruptcy.** *"A small UK company's daily stock price in GBP over 50 trading days. On day 30 the company enters compulsory liquidation and is permanently delisted from the London Stock Exchange."* squared exponential kernel ($\kappa = 5/50$, $\tau^2 = 2.0$), output bounded to $[0, 10]$.
- **Stable.** *"A small UK company's daily stock price in GBP over 50 trading days. The company has no notable news during the period, but performs well. Ending with a value over 5 GBP per share."* Same squared exponential kernel as above, output bounded to $[0, 10]$.
- **San Diego (precipitation).** *"Monthly average precipitation in San Diego in inches over 50 months, starting in January."* Product of a squared exponential kernel and a periodic kernel (period $= 12/50$, $\kappa = 1.0$, $\tau^2 = 2.0$), output bounded to $[0, 4]$.

All three scenarios use $m = 50$ output time steps, 100 predictive samples, two weak anchor observations at $t \in \{1/50, 2/50\}$ to initialise the GP posterior, and run for $1{,}000$ denoising steps with 1 MC sample per step in the whitened representation.

K Ablation Study: Guidance Method and Discretisation

We ablate two axes of our method on the monotonicity-and-boundedness task in Figure 6: (i) the guidance estimator and (ii) the number of denoising steps T. All combinations are evaluated qualitatively by inspecting whether $S = 1{,}000$ posterior samples respect the monotonicity and boundedness constraints and display appropriate posterior variation.

Guidance estimators. We compare four estimators for the conditional score term $\nabla_{\mathbf{f}_t} \log p(\mathcal{C} \mid \mathbf{f}_t)$:

- **DPS** [43] estimates the guidance term with first-order approximations $\nabla_{\boldsymbol{f}_t} \log p(\mathcal{C} \mid \mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t])$, where $\mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t] = \frac{1}{\alpha(t)} \big(\boldsymbol{f}_t + (1 - \alpha(t)^2) s(\boldsymbol{f}_t, t)\big)$.
- **MPGD** [44] uses $\nabla_{\boldsymbol{f}_0} \log p(\mathcal{C} \mid \mathbb{E}[\boldsymbol{f}_0 \mid \boldsymbol{f}_t])$, avoiding differentiation through the denoising process.
- **MC (ours).** Importance-weighted MC gradient using $S \in \{1, 10, 100\}$ samples as described in Appendix F.
- **MC-Fisher (ours).** An alternative MC estimator that forms the guidance direction as the importance-weighted deviation of the denoised samples from their mean, using $S \in \{1, 10, 100, 1000\}$

Non-MC approaches underrepresent predictive variance. Figure 6 shows that both DPS and MPGD produce samples that collapse to near-deterministic trajectories, dramatically underrepresenting the true posterior uncertainty. Because both methods condition on a single point estimate of $\boldsymbol{f}_0$ rather than integrating over the denoising predictive distribution, the guidance signal is overconfident and pushes all samples towards a narrow mode. The resulting sample sets show almost no spread, failing to capture the breadth of constraint-satisfying functions consistent with the observations.

Our MC estimator works well even with a single sample. The importance-weighted MC estimator produces diverse, well-constrained samples already at $S = 1$. Even a single draw from the denoising predictive distribution is sufficient to obtain an unbiased direction that preserves predictive variance, and quality improves gracefully as S increases. This makes the MC estimator both practical and accurate in the regime relevant to our experiments.

MC-Fisher requires many samples. The Fisher-based MC estimator is substantially less sample-efficient. With $S < 1000$ the resulting samples still collapse and exhibit poor diversity, and even at $S = 1000$ the sample spread is only beginning to look reasonable. We therefore favour the standard MC estimator in all main experiments.

Number of steps. With $T = 10$ steps the ODE discretisation is too coarse and constraint violations are common regardless of estimator. At $T = 100$ the MC estimator begins to produce satisfactory results. $T = 1000$ steps consistently yields smooth, well-constrained samples under our MC estimator and this choice is used in all main experiments, unless otherwise stated.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions and scope. The four bullet-point contributions in Section 1 correspond directly to theoretical results (Theorems D.1, D.3, E.1) and experimental demonstrations (Sections 6.1–6.3, Appendix I).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed explicitly in Section 7, covering three key areas: (i) cubic scaling in mm and the restriction to finite-dimensional discretisations; (ii) theoretical guarantees limited to log-concave likelihoods, with only worst-case exponential bounds for non-log-concave settings such as LLM-based likelihoods; and (iii) systematic bias in hyperparameter estimation when the non-Gaussian condition CC is highly informative. The practical concern of importance weight collapse in high dimensions is acknowledged in Section 4.2, and the restriction to the VP-SDE formulation is noted in Section 3.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

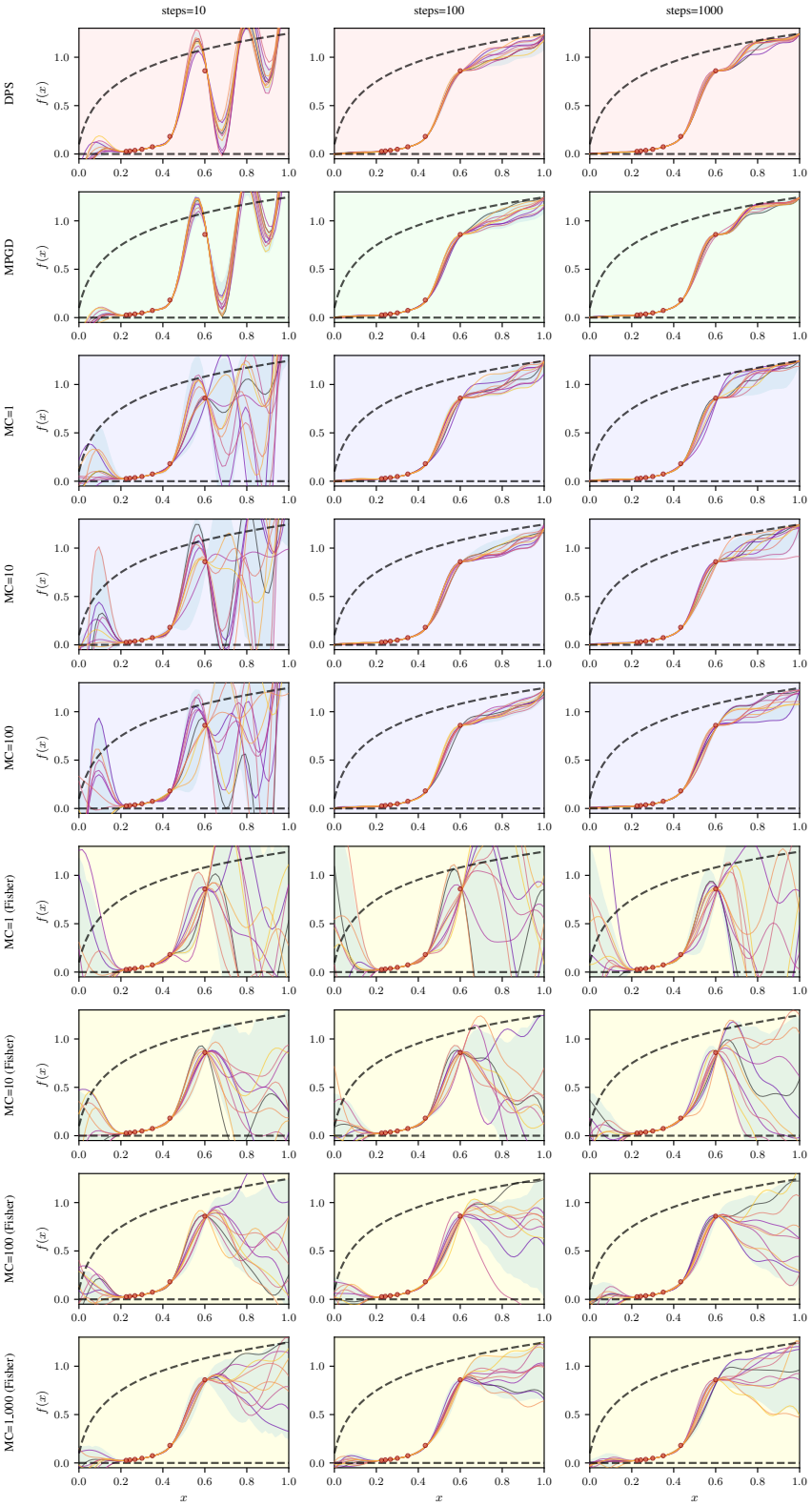


Figure 6: Additional experiments on the monotonic and bounded regression problem, showing the effect of different time discretisations (left to right), for guidance estimation strategies (top to bottom), including DPS (red), MPGD (green), our MC estimator (Blue), and our MC-fisher estimator (Yellow).

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are stated with full assumptions and complete proofs. Theorem D.1 (stiffness equivalence), Theorem D.3 and Corollary D.4 (Wasserstein-2 optimality of whitening), and Theorem E.1 (stability of guided sampling) are proved in full in Appendices D and E respectively, with assumptions stated explicitly in each theorem statement. Proposition E.5 and its corollaries are proved in Appendix E.1–E.2. All theorems and lemmas are numbered and cross-referenced throughout the main text and appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full experimental details are provided in Appendices G–K, including data generation procedures, GP prior specifications, kernel hyperparameters and how they are selected, ODE solver settings, number of Monte Carlo samples, and train/test splits. Implementation details common to all experiments are given in Section 6 and Appendix F, including the diffusion schedule, time discretisation, and gradient clipping procedure. Baseline comparisons use the same data splits and evaluation metrics (RMSE and NLPD) as the original papers where applicable, and deviations are noted explicitly (e.g. hyperparameter fitting procedures in Section 6.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided at the anonymised repository linked in Section 6. The repository includes implementations of FLOWGP and all baselines, scripts to reproduce all figures and tables in the main paper. Data for the physics experiments is taken from the publicly available PHYSS [36] benchmark; LLM-derived densities are generated using the code of Biggs et al. [57].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details are specified in Appendices G-K for each experiment, including data splits, kernel hyperparameters and how they are chosen (marginal likelihood maximisation) ODE solver settings (Euler scheme, SNR uniform discretisation), number of Monte Carlo samples, and likelihood parameters for each constraint type. Hardware is specified in Section 6 (Apple M2 Pro for 1D experiments, Nvidia A4500 for 2D physics experiments). Baseline hyperparameters follow the original papers where applicable, with any deviations noted in Section 6.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We include bootstrap-based error bars for our table-based results and predictive confidence intervals in the appropriate figures (including the BO example).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Hardware is specified in Section 6.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and we conform with this in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Section 7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: The paper does not release pre-trained language models, image generators, or scraped datasets. The LLM component uses Qwen3.5 as a black-box inference tool and does not involve fine-tuning or releasing model weights. The code release contains only GP inference routines and experiment scripts, which pose no meaningful risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets are properly credited. The PHYSS benchmark data is sourced from Hamelijnck et al. [31] and the AUTOIP benchmark from Long et al. [32]; both are cited appropriately. The LLM-derived densities are generated using the code of Biggs et al. [49], which is cited and credited. The PairwiseGP model is from BoTorch [60], which is cited. The Qwen3.5 LLM [48] is cited with its technical report. Baseline implementations follow the original papers in all cases.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The primary new asset introduced is the FLOWGP codebase, provided at the anonymised repository linked in Section 6. No new datasets are introduced; all experimental data is either synthetically generated (physics experiments) or sourced from existing benchmarks as described in Appendices G–K. The repository will be de-anonymised and assigned an open-source licence upon acceptance.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on synthetic data, existing physics benchmarks, or LLM-generated probabilistic predictions over time series. No human participants were involved at any stage of the research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, and therefore no IRB approval or equivalent ethical review was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

Justification: The use of a large language model is an important and original component of the core methodology in Section 6.3. Specifically, Qwen3.5 [56] is used to generate marginal predictive densities over time series conditioned on natural language prompts, which are then incorporated into the FLOWGP framework via a product-of-experts construction. Full details of the LLM integration, including the smoothing procedure and the product-of-experts formulation, are given in Section 6.3 and Appendix J. We follow the setup of Biggs et al. [57] for the LLM-derived densities and refer to their paper for details of the prompting strategy, temperature settings, and post-processing

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.