Scalable Gaussian Process Flows

Thomas Cowperthwaite
University of Cambridge
tc656@cam.ac.uk

Lachlan Astfalck
University of New South Wales

Henry Moss
Lancaster University

Louis Sharrock
University College London

Abstract

Outside of the linear-Gaussian regime, conditional sampling from Gaussian processes (GPs) is challenging. Recent sampling algorithms provide machinery allowing conditioning on arbitrary statements, including non-linear and non-Gaussian conditions. However, important bottlenecks remain: typically the additional flexibility comes at an increased computational cost. This is seen prominently in [1], which vastly increases the scope of GP modelling capabilities, but at the cost of executing an expensive iterative diffusion model. In this paper, we alleviate two significant drawbacks of [1] by (1) introducing kernel approximations that improve scalability to high-resolution domains and (2) allowing for kernel hyperparameter optimisation. We demonstrate the utility on two important downstream tasks: probabilistic downsampling using areal summary statistics, and inference of PDE solutions on irregular domains from noisy observations.

1 Introduction

Gaussian processes (GPs) provide flexible probabilistic models over unknown functions, with calibrated uncertainty and a natural means of incorporating prior knowledge [2]. Conditioning on data is analytically tractable for linear observations with Gaussian noise; however, scientific information often takes more complex forms, including non-linear differential equations, integral measurements, inequalities, and boundary constraints. Existing solutions typically rely on posterior approximations such as Laplace [3], expectation propagation [4], variational inference [5], or on specialised kernels and inference schemes derived for a particular class of constraints [6, 7]. Recently, FLOWGP [1] introduced a general framework for conditioning GPs on *non-linear* and *non-Gaussian* conditions.

FLOWGP operates over a fixed discretisation of the function's domain, and thus the computational requirements grow with the number of evaluation points N, including an initial $\mathcal{O}(N^3)$ matrix factorisation and an N-dimensional ODE solve. This becomes restrictive for fine spatial or spatio-temporal grids. The factorisation step also makes repeated computation across kernel hyperparameters prohibitively expensive for all but the smallest-scale problems. Finally, known structure such as boundary conditions must typically be represented through additional conditioning statements or artificial observations, which either makes the sampling ODE numerically challenging to solve, or introduces additional computational cost [1] that we may otherwise avoid.

In this work, we extend FLOWGP to operate on the weights of a finite-feature GP approximation instead of function values, thus improving scalability and flexibility. The resulting flow has dimension $R \ll N$, the (tunable) number of features, and decouples the cost of the flow from the resolution at which its function samples are evaluated. This makes kernel hyperparameter optimisation practical and the feature representation also provides access to analytic derivatives which may be further exploited. We evaluate the resulting method through wall-time scaling and numerical-accuracy, across two applications: probabilistic downscaling [8, 9] and inference of PDE solutions on irregular

Preprint.

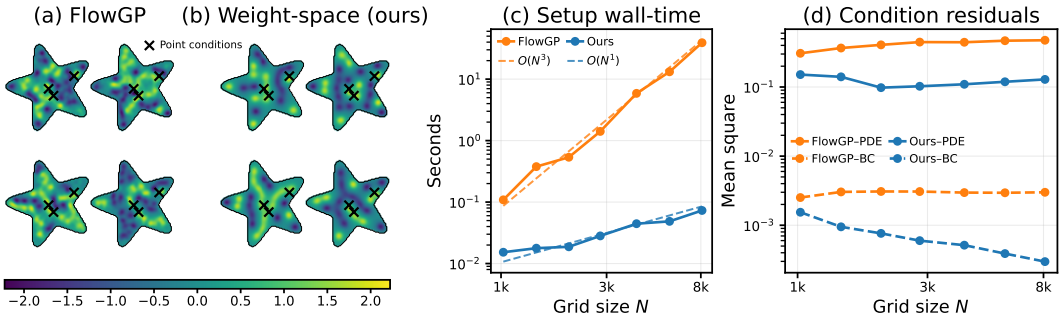

Figure 1: Gaussian process samples, conditioned on non-conjugate information (satisfying a non-linear PDE). **(a)** samples using FLOWGP, using artificial data to enforce boundary conditions. **(b)** samples drawn using our new weight-space adaptation, boundary conditions are satisfied by construction. **(c)** comparison of setup wall-time as a function of grid resolution. **(d)** our samples are of better quality compared to those using FLOWGP, as measured by the PDE residual.

domains [10, 11]. Our results extend general-purpose conditional GP sampling to settings in which a dense function-space formulation like FLOWGP is computationally impractical (see Fig. 1).

2 Gaussian Processes and Sampling

Let $f\colon \mathcal{X} \to \mathbb{R}$ be a random function over $\mathcal{X} \subseteq \mathbb{R}^D$. Assume f has a GP prior, written $f \sim \mathcal{GP}(\mu, k_{\boldsymbol{\theta}})$, specified by a mean function $\mu\colon \mathcal{X} \to \mathbb{R}$ and a positive semidefinite kernel $k_{\boldsymbol{\theta}}\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with hyperparameters $\boldsymbol{\theta}$. By definition, for every finite collection of inputs $X = (x_1, \ldots, x_N)$, the vector of GP function evaluations $\mathbf{f} \coloneqq [f(x_1), \ldots, f(x_N)]^\top$ is multivariate Gaussian distributed $\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}(\boldsymbol{\theta}))$, with elements $\mathbf{m}_i = \mu(x_i)$ and $\mathbf{K}_{ij}(\boldsymbol{\theta}) = k_{\boldsymbol{\theta}}(x_i, x_j)$ for $i, j \in \{1, \ldots, N\}$.

To sample from the GP at X, one can first draw white noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$, and then apply the mean-scale transformation $\mathbf{f} = \mathbf{m} + \mathbf{L}\mathbf{z}$, where $\mathbf{L}\mathbf{L}^\top = \mathbf{K}$. In general, constructing $\mathbf{L}$ costs $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ memory, and the subsequent sample costs $\mathcal{O}(N^2)$. Consequently, sampling functions on a fine spatial grid becomes prohibitively expensive as N grows.

To scale to fine domain discretisations, one can approximate the kernel using basis functions $\{\psi^r_{\boldsymbol{\theta}}\}_{r=1}^R$ with feature vector $\boldsymbol{\psi}_{\boldsymbol{\theta}}(x) = [\psi^1_{\boldsymbol{\theta}}(x), \ldots, \psi^R_{\boldsymbol{\theta}}(x)]^\top$ and matrix $\boldsymbol{\Psi}_{\boldsymbol{\theta}} = \boldsymbol{\psi}_{\boldsymbol{\theta}}(X)^\top \in \mathbb{R}^{N \times R}$. For a suitable choice of basis representation (cf. Sec. A for some examples), the kernel approximation takes the form $k_{\boldsymbol{\theta}}(x, x') \approx \boldsymbol{\psi}_{\boldsymbol{\theta}}(x)^\top \boldsymbol{\psi}_{\boldsymbol{\theta}}(x')$, so that $\mathbf{K}(\boldsymbol{\theta}) \approx \widetilde{\mathbf{K}}(\boldsymbol{\theta}) = \boldsymbol{\Psi}_{\boldsymbol{\theta}} \boldsymbol{\Psi}_{\boldsymbol{\theta}}{}^\top$. Equivalently, the approximate GP may be represented directly in weight space using the decoder function $g_{\boldsymbol{\theta}}$, namely,

$$g_{\boldsymbol{\theta}}(\boldsymbol{\xi}) \coloneqq \widetilde{f}_{\boldsymbol{\theta}} = \mu + \boldsymbol{\psi}_{\boldsymbol{\theta}}^\top \boldsymbol{\xi}, \qquad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R). \tag{1}$$

A functional sample can be evaluated at arbitrary locations, incurring only a $\mathcal{O}(NR)$ cost [12, 13].

Finite-feature representations do not resolve the difficulty of conditioning on non-conjugate information. FLOWGP, introduced in [1] and since extended in [14], provides a general framework for this task, by reframing conditional sampling as a transport problem. Consider the Gaussian interpolation

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

where $\alpha(t) = \exp(-\frac{1}{2}\int_0^t \beta(z)\mathrm{d}z)$. As t increases, this progressively replaces the sample $\mathbf{f}_0$ with Gaussian noise, defining a continuous path from the GP distribution to a standard Gaussian reference. Since $\mathbf{f}_0 \sim \mathcal{N}(\mathbf{m}, \mathbf{K}(\boldsymbol{\theta}))$, the marginals of this interpolation are given in closed form:

$$p_t(\mathbf{f}_t) = \mathcal{N}\big(\alpha(t)\mathbf{m},\, \alpha(t)^2\mathbf{K}(\boldsymbol{\theta}) + \big(1 - \alpha(t)^2\big)\mathbf{I}_N\big) \tag{3}$$

Samples can be transported along this path in the opposite direction by simulating the corresponding reverse-time stochastic differential equation (SDE)

$$\mathrm{d}\mathbf{f}_t = \beta(t)\big[\tfrac{1}{2}\mathbf{f}_t - \nabla_{\mathbf{f}} \log p_t(\mathbf{f}_t)\big]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\overline{\mathbf{W}}_t, \qquad t : 1 \to 0, \tag{4}$$

where $\overline{\mathbf{W}}_t$ is a reverse-time Wiener process [15]. Crucially, because p_t is Gaussian, the unconditional score $\nabla_{\mathbf{f}_t} \log p_t(\mathbf{f}_t)$ is available analytically. Conditioning can now be incorporated by augmenting Eq. (4) with a likelihood-dependent guidance term: for information $\mathcal{C}$ and by Bayes' rule

$$\nabla_{\mathbf{f}_t} \log p_t(\mathbf{f}_t \mid \mathcal{C}) = \underbrace{\nabla_{\mathbf{f}_t} \log p_t(\mathbf{f}_t)}_{\text{closed-form GP score}} + \underbrace{\nabla_{\mathbf{f}_t} \log p_t(\mathcal{C} \mid \mathbf{f}_t)}_{\text{conditional guidance}}. \tag{5}$$

The second term can be approximated using samples from $p(\mathbf{f}_0 \mid \mathbf{f}_t)$, requiring only pointwise evaluation of a likelihood function $p(\mathcal{C} \mid \mathbf{f}_0)$. This allows the sampler to accommodate non-linear and non-Gaussian observations, structural restrictions, and other black-box conditioning statements.

3 Methods

We address the limitations presented in Sec. 2 by simulating the SDE in Eq. (4) in the coefficient space of a finite GP expansion described in Eq. (1). This separates the dimension of the sampling dynamics from the resolution of the evaluation grid, making repeated computation possible. In turn, this enables scalable estimation of kernel hyperparameters. We introduce methodology to adapt the kernel hyperparameters during integration using the same conditional information that guides the flow, resulting in a parsimonious framework for joint estimation of the process f and its hyperparameters.

Conditional flow in weight space. Using an appropriate kernel approximation, we apply the same variance-preserving interpolation as in function space directly to the weight vector,

$$\boldsymbol{\xi}_t = \alpha(t)\boldsymbol{\xi}_0 + \sqrt{1-\alpha(t)^2}\,\mathbf{z}, \qquad \boldsymbol{\xi}_0, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R). \tag{6}$$

The probability-flow SDE consequently has dimension R rather than N. Guidance is evaluated in weight space: for a condition $\mathcal{C}$, its likelihood is computed after applying the differentiable decoder in Eq. (1). This highly-scalable flow over approximate GPs via their weight-space representation is a special case of a tilt as given in [14]. Function values are formed on the full output grid only when required by a condition or when returning the final sample. A condition may instead be evaluated at a smaller set of collocation points, making its cost independent of the resolution used for display.

As the flow is performed in weight space, the major cost of sampling, namely kernel factorisation, is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(R^3)$ with $R \ll N$. Evaluating the decoder on N_c locations therefore costs $\mathcal{O}(RN_c)$; with conjugate data, the remaining dense factorisation is only dependent on the number of observed data (cf. Sec. A.3). The cost of adaptive selection is thus governed by the number of features and the number of observations, rather than by the resolution of the output grid. For a stationary prior encoded by a differentiable basis, derivatives of samples may be computed analytically without the need for a finite-difference scheme as $\partial_{x_d} \widetilde{f}(x) = \partial_{x_d}\mu(x) + (\partial_{x_d}\boldsymbol{\psi}(x))^\top \boldsymbol{\xi}$. Differential equation residuals can therefore be evaluated in $\mathcal{O}(RN_c)$ time.

Hyperparameter optimisation. For conjugate data, the parameters of the prior mean and covariance can be adjusted by maximising the marginal log-likelihood of the observations. In principle, we could learn the hyperparameters in the same way for FLOWGP. However, repeatedly running the conditional sampler within a hyperparameter optimiser would be computationally expensive. Moreover, parameter estimates would then be independent of the informative non-conjugate conditions.

Instead, we propose to adapt the hyperparameters using the same information that guides the sampler. The natural objective is the marginal likelihood, or evidence, of the condition. This is given by

$$\begin{aligned} J(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathcal{C}) &= \log \mathbb{E}_{\boldsymbol{\xi}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R)} \left[p(\mathcal{C} \mid g_{\boldsymbol{\theta}}(\boldsymbol{\xi}_0)) \right] \\ &= \log \mathbb{E}_{\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R)} \left[\mathbb{E}_{\boldsymbol{\xi}_0 \sim p(\boldsymbol{\xi}_0 \mid \boldsymbol{\xi}_t)} \left[p(\mathcal{C} \mid g_{\boldsymbol{\theta}}(\boldsymbol{\xi}_0)) \right] \right], \end{aligned} \tag{7}$$

where final equality follows from the law of total expectation and the fact that Eq. (6) preserves the standard Gaussian marginal. If $\mathcal{C}$ contains a linear-Gaussian component, part of this objective can be evaluated analytically; see Appendix B. The inner expectation above is precisely the smoothed condition likelihood already used to guide the conditional flow. Thus, quantities required to optimise the evidence are already evaluated during sampling.

During numerical integration, we can exploit Eq. (7) to update the kernel hyperparameters every q SDE steps. To be specific, for a given update time t, we hold the current weight-space states fixed and, for each of a batch of B trajectories, draw S reparameterised terminal estimates $\{\boldsymbol{\xi}_0^{(b,s)}\}_{s=1}^{S}$

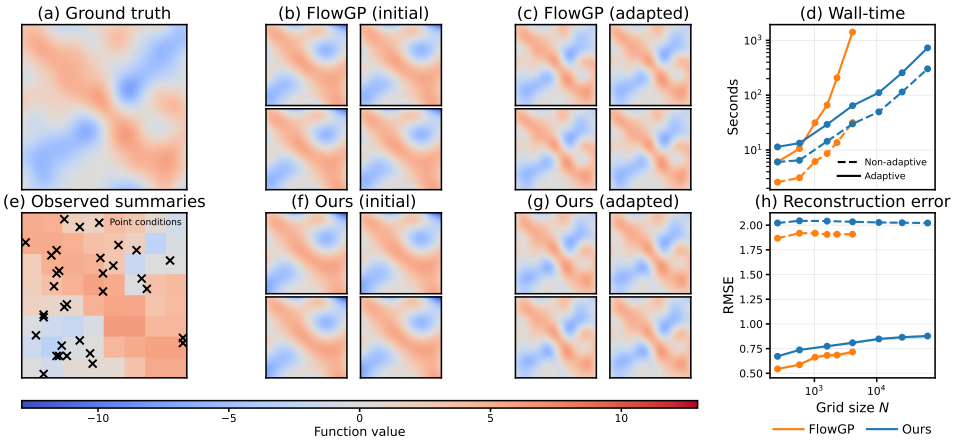

Figure 2: A non-linear downscaling task. **(a)** the ground truth field. **(e)** areal summary statistics on a coarse grid, with pointwise conjugate conditioning data shown. **(b, f)** samples from the conditioned flows with a deliberately misspecified kernel. **(c, g)** draws from the flows conducted with fitted kernel hyperparameters. **(d, h)** comparison of FLOWGP and our weight-space method in adaptive and non-adaptive modes, showing variation of total wall time and RMSE with grid resolution.

from $p(\boldsymbol{\xi}_0 \mid \boldsymbol{\xi}_t^{(b)})$. Replacing the inner expectation in Eq. (7) by its Monte Carlo approximation gives

$$\widehat{J}_t(\boldsymbol{\theta}) = \frac{1}{B}\sum_{b=1}^{B} \log\left[\frac{1}{S}\sum_{s=1}^{S} p\left(\mathcal{C} \mid g_{\boldsymbol{\theta}}(\boldsymbol{\xi}_0^{(b,s)})\right)\right]. \tag{8}$$

The resulting stochastic gradients can be computed directly using automatic differentiation, and then used within a general purpose optimiser (e.g., Adam) to optimise the objective.

4 Results

Probabilistic downscaling of spatial fields is an increasingly important use case for generative modelling [8, 16–18]. The fundamental task is to produce probable, high-resolution fields, given some summary statistic measured on a coarser grid, often alongside direct measurements or prior knowledge (e.g. imposed physics).

We first generate a high-resolution ground-truth field (Fig. 2a) by sampling an approximate posterior GP with a large number of random Fourier features (RFFs) ($R = 4096$). This field is then coarse-grained into an 8×8 grid using the (non-linear) areal $\max$ summary statistic to constitute $\mathcal{C}$, and 32 random locations in the domain are noisily queried to define a linear condition (Fig. 2e). We then generate conditional samples using both FLOWGP and our RFF weight-space adaptation with deliberately misspecified kernel hyperparameters (Figs. 2b, f), resulting in poor quality samples. The experiment is then repeated using the adaptive sampling scheme, resulting in adjusted hyperparameters (Fig. 3) and higher-quality samples, as measured by RMSE (Figs. 2c, g, h). While reconstruction RMSE increases by 12% with the weight-space flow, FLOWGP quickly becomes infeasibly computationally expensive, as measured by total wall-time (Fig. 2d), our method is 22$\times$ faster. Further details are found in Sec. C.1.

PDE solutions are commonly subject to boundary conditions (BCs), whose type and sufficiency determine the uniqueness of a solution [19]. In the dense FLOWGP framework, BCs can only be enforced inelegantly by conditioning on additional artificial noiseless data. Here, we leverage the work of Solin and Kok [20] to use the Laplacian eigenfunction basis to enforce homogeneous Dirichlet BCs beyond the linear-Gaussian regime they demonstrate.

We show GP samples from both FLOWGP (Fig. 1a) and our weight-space (Fig. 1b) flow, conditioned such that they solve the Allen-Cahn equation: $f = f^3 + 10^{-3} \cdot \Delta f$ on the irregular domain $\Omega \subseteq \mathbb{R}^2$, where Δ is the Laplacian, by minimising the associated residual. We impose the Dirichlet BC $f = 0$ on $\partial\Omega$ using artificial data for FLOWGP, and via an appropriate Laplacian basis for the weight-space flow (Sec. A.2). We again note that the weight-space setup wall-time empirically scales much more

favourably with domain resolution than the exact flow (Fig. 1c), while achieving better adherence to the BC and a 72.9% improvement in mean squared PDE residual (Fig. 1d). Further details are found in Sec. C.2.

Conclusion. We demonstrate that a weight-space parameterisation of flow-based, non-conjugate GP sampling vastly improves scalability, allowing for principled hyperparameter optimisation, with only modest decreases in sample quality on some tasks. Future work will pursue newly unlocked real-world applications, extend empirical validation, and strengthen theoretical guarantees.

References

[1] Henry Moss, Lachlan Astfalck, Thomas Cowperthwaite, Colin Doumont, Sam Willis, Philipp Hennig, Christopher Nemeth, and Andrew Zammit-Mangion. Conditioning Gaussian Processes on Almost Anything, May 2026. URL http://arxiv.org/abs/2605.21041. arXiv:2605.21041 [stat.ML].

[2] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9. OCLC: ocm61285753.

[3] 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.

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

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

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

[7] Da Long, Zheng Wang, Aditi Krishnapriyan, Robert Kirby, Shandian Zhe, and Michael Mahoney. AutoIP: A United Framework to Integrate Physics into Gaussian Processes. In *Proceedings of the 39th International Conference on Machine Learning*, pages 14210–14222. PMLR, June 2022. URL https://proceedings.mlr.press/v162/long22a.html.

[8] Pavel Perezhogin, Laure Zanna, and Carlos Fernandez-Granda. Generative Data-Driven Approaches for Stochastic Subgrid Parameterizations in an Idealized Ocean Model. *Journal of Advances in Modeling Earth Systems*, 15(10):e2023MS003681, 2023. ISSN 1942-2466. doi: 10.1029/2023MS003681. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2023MS003681. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2023MS003681.

[9] Jonathan Schmidt, Luca Schmidt, Felix M. Strnad, Nicole Ludwig, and Philipp Hennig. A Generative Framework for Probabilistic, Spatiotemporally Coherent Downscaling of Climate Simulation. *npj Climate and Atmospheric Science*, 8(1):270, July 2025. ISSN 2397-3722. doi: 10.1038/s41612-025-01157-y. URL https://www.nature.com/articles/s41612-025-01157-y.

[10] Biswajit Khara, Ethan Herron, Zhanhong Jiang, Aditya Balu, Chih-Hsuan Yang, Kumar Saurabh, Anushrut Jignasu, Soumik Sarkar, Chinmay Hegde, Adarsh Krishnamurthy, and Baskar Ganapathysubramanian. Neural PDE Solvers for Irregular Domains, November 2022. URL http://arxiv.org/abs/2211.03241. arXiv:2211.03241 [cs.LG].

[11] Zhiwei Zhao, Changqing Liu, Yingguang Li, Zhibin Chen, and Xu Liu. Diffeomorphism neural operator for various domains and parameters of partial differential equations. *Communications Physics*, 8(1):15, January 2025. ISSN 2399-3650. doi: 10.1038/s42005-024-01911-3. URL https://www.nature.com/articles/s42005-024-01911-3.

[12] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html.

[13] Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, March 2020. ISSN 1573-1375. doi: 10.1007/s11222-019-09886-w. URL https://doi.org/10.1007/s11222-019-09886-w.

[14] Louis Sharrock, Lachlan Astfalck, and Henry Moss. LatentFlow: A General Framework for Conditioning Stochastic Processes, July 2026. URL http://arxiv.org/abs/2607.12922. arXiv:2607.12922 [stat.ML].

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

[16] Philipp Hess, Michael Aich, Baoxiang Pan, and Niklas Boers. Fast, scale-adaptive and uncertainty-aware downscaling of Earth system model fields with generative machine learning. *Nature Machine Intelligence*, 7(3):363–373, March 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-00980-5. URL https://www.nature.com/articles/s42256-025-00980-5.

[17] Jordan P. Deshon, Jeffrey D. Niemann, Timothy R. Green, Andrew S. Jones, and Peter J. Grazaitis. Stochastic analysis and probabilistic downscaling of soil moisture in small catchments. *Journal of Hydrology*, 585:124711, June 2020. ISSN 0022-1694. doi: 10.1016/j.jhydrol.2020.124711. URL https://www.sciencedirect.com/science/article/pii/S0022169420301712.

[18] Nadav Peleg, Simone Fatichi, Athanasios Paschalis, Peter Molnar, and Paolo Burlando. An advanced stochastic weather generator for simulating 2-D high-resolution climate variables. *Journal of Advances in Modeling Earth Systems*, 9(3):1595–1627, 2017. ISSN 1942-2466. doi: 10.1002/2016MS000854. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/2016MS000854.

[19] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010. ISBN 9780821849743 0821849743.

[20] Arno Solin and Manon Kok. Know Your Boundaries: Constraining Gaussian Processes by Variational Harmonic Features. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 2193–2202. PMLR, April 2019. URL https://proceedings.mlr.press/v89/solin19a.html.

[21] James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, August 2020. URL https://arxiv.org/abs/2002.09309.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs].

A Basis Decompositions

A.1 Random Fourier features

Random Fourier features (RFFs) provide a finite-dimensional approximation to stationary kernels [12]. By Bochner's theorem, a stationary kernel $k_{\boldsymbol{\theta}}(x, x') = k_{\boldsymbol{\theta}}(x - x')$ can be written as an expectation over its spectral density $p_{\boldsymbol{\theta}}(\boldsymbol{\omega})$. Drawing frequencies $\boldsymbol{\omega}_r \sim p_{\boldsymbol{\theta}}$ and phases $b_r \sim \mathrm{Uniform}(0, 2\pi)$, we define

$$\phi_{\boldsymbol{\theta}}^{(r)}(x) = \sqrt{\frac{2\sigma_f^2}{R_f}} \cos\left(\boldsymbol{\omega}_r^\top x + b_r\right), \qquad r = 1, \ldots, R_f \tag{9}$$

With R_f defined as the number of features used to approximate the kernel. The resulting feature vector satisfies

$$k_{\boldsymbol{\theta}}(x, x') \approx \boldsymbol{\phi}_{\boldsymbol{\theta}}(x)^\top \boldsymbol{\phi}_{\boldsymbol{\theta}}(x'), \tag{10}$$

and an approximate GP prior draw is

$$\widetilde{f}_{\boldsymbol{\theta}}(x) = \mu(x) + \boldsymbol{\phi}_{\boldsymbol{\theta}}(x)^\top \boldsymbol{w}, \qquad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{R_f}). \tag{11}$$

For the RBF kernel with length-scale matrix $\boldsymbol{\Lambda}_{\boldsymbol{\theta}}$, the spectral density is Gaussian,

$$\boldsymbol{\omega}_r \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}\right). \tag{12}$$

Fixing the underlying standard-normal frequency draws while varying $\boldsymbol{\theta}$ gives a differentiable parameterisation of the features. Spatial derivatives are also available analytically by differentiating the trigonometric basis. Although RFFs directly approximate only the stationary prior kernel, they can be combined with exact kernel evaluations in the decoupled posterior construction of Section A.3.

A.2 Laplacian eigenfunction basis

Laplacian eigenfunctions provide a deterministic low-rank approximation to stationary kernels while allowing boundary conditions to be imposed on a bounded domain $\Omega \subset \mathbb{R}^d$ [20]. For homogeneous Dirichlet boundary conditions, the basis functions are obtained from the eigenvalue problem

$$-\Delta\varphi_j(x) = \lambda_j^2 \varphi_j(x), \qquad x \in \Omega, \qquad \varphi_j(x) = 0, \qquad x \in \partial\Omega, \tag{13}$$

where the eigenfunctions are orthonormal in $L^2(\Omega)$ and ordered by increasing eigenvalue.

Let $s_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ denote the spectral density of the stationary kernel. For an isotropic kernel, its value depends only on the radial frequency, and the covariance function can be approximated using the first R_f eigenfunctions as

$$k_{\boldsymbol{\theta}}(x, x') \approx \sum_{j=1}^{R_f} s_{\boldsymbol{\theta}}(\lambda_j)\varphi_j(x)\varphi_j(x'). \tag{14}$$

Defining the weighted features

$$\phi_{\boldsymbol{\theta}}^{(j)}(x) = \sqrt{s_{\boldsymbol{\theta}}(\lambda_j)}\,\varphi_j(x), \qquad j = 1, \ldots, R_f, \tag{15}$$

gives

$$k_{\boldsymbol{\theta}}(x, x') \approx \boldsymbol{\phi}_{\boldsymbol{\theta}}(x)^\top \boldsymbol{\phi}_{\boldsymbol{\theta}}(x'), \tag{16}$$

and the corresponding approximate prior draw is

$$\widetilde{f}_{\boldsymbol{\theta}}(x) = \mu(x) + \boldsymbol{\phi}_{\boldsymbol{\theta}}(x)^\top \boldsymbol{w}, \qquad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{R_f}). \tag{17}$$

For simple domains, such as intervals and rectangles, the eigenpairs are available analytically. For irregular domains, they can instead be computed once by discretising the Laplace operator on a grid and solving the resulting sparse eigenvalue problem. Restricting the discretisation to points inside Ω naturally imposes the boundary condition and ensures that every approximate prior draw vanishes on $\partial\Omega$. Other conditions, including homogeneous Neumann conditions, can be incorporated by changing the boundary condition in the eigenvalue problem.

The eigenfunctions and eigenvalues depend only on the domain and can therefore be held fixed while varying $\boldsymbol{\theta}$; the kernel hyperparameters enter only through the spectral weights $s_{\boldsymbol{\theta}}(\lambda_j)$. This gives a differentiable parameterisation with respect to the kernel hyperparameters, while replacing dense kernel operations by computations involving R_f basis functions.

A.3 Decoupled posterior basis

Naïve kernel approximation cannot be applied directly to a GP posterior, because conditioning destroys stationarity. Given Gaussian observations $\mathcal{D} = (X_y, \mathbf{y})$ with noise variance σ_y^2, we instead use a decoupled pathwise representation [21]. Let

$$\mathbf{A}_{\boldsymbol{\theta}} = \left(\widetilde{\mathbf{K}}(\boldsymbol{\theta}) + \sigma_y^2 \mathbf{I}\right)^{-1}, \qquad \widetilde{\mathbf{K}}(\boldsymbol{\theta}) = \boldsymbol{\Phi}_{\boldsymbol{\theta}} \boldsymbol{\Phi}_{\boldsymbol{\theta}}^\top, \tag{18}$$

where the feature matrix $\boldsymbol{\Phi}_{\boldsymbol{\theta}} = \boldsymbol{\phi}_{\boldsymbol{\theta}}(X)^\top \in \mathbb{R}^{N \times R_f}$ may use any stationary kernel decomposition, including those in Sections A.1 and A.2. Matheron's update applied to an approximate prior draw can be collected into the affine form

$$g_{\boldsymbol{\theta}}(\boldsymbol{\xi}) \coloneqq \widetilde{f}_{\boldsymbol{\theta}} = \mu + \boldsymbol{\psi}_{\boldsymbol{\theta}}^\top \boldsymbol{\xi}, \qquad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R), \tag{19}$$

with

$$\boldsymbol{\psi}_{\boldsymbol{\theta}}(x) = \begin{bmatrix} \boldsymbol{\phi}_{\boldsymbol{\theta}}(x) - \boldsymbol{\Phi}_{\boldsymbol{\theta}} \mathbf{A}_{\boldsymbol{\theta}} k_{\boldsymbol{\theta}}(X_y, x) \\ -\sigma_y \mathbf{A}_{\boldsymbol{\theta}} k_{\boldsymbol{\theta}}(X_y, x) \end{bmatrix}. \tag{20}$$

The second block is omitted for noiseless observations. This construction approximates only the stationary prior, while applying the rank-$|X_y|$ conditioning correction with the canonical kernel. Its latent dimension is $R = R_f + |X_y|$ (where R_f is the rank of the prior approximation) when $\sigma_y^2 > 0$, and its mean is the exact GP posterior mean. For the RBF kernel, derivatives of both the features and posterior mean are computed analytically; derivatives of the canonical-kernel correction use the corresponding Hermite-polynomial factors.

B Hyperparameter Optimisation

In Sec. 3, we note that when the conditioning statement $\mathcal{C}$ contains some linear-Gaussian component, we can extract this part of the likelihood and compute it analytically. In this case, for conjugate conditioning statement $\mathcal{D}$ (e.g. noisy point observations with Gaussian likelihood), we split the objective function into two terms:

$$\widehat{J}_l(\boldsymbol{\theta}) = \frac{1}{B} \sum_{b=1}^{B} \log \left[\frac{1}{S} \sum_{s=1}^{S} p\Big(\mathcal{C} \mid \Big(g_{\boldsymbol{\theta}}(\boldsymbol{\xi}_0^{(b,s)}) \mid \mathcal{D}\Big)\Big) \right] + \log p(\mathcal{D} \mid \boldsymbol{\theta}). \tag{21}$$

where the second term may be computed without Monte Carlo (MC) sampling.

For the particular case in which $\mathcal{D} = (X_y, \mathbf{y})$ consists of pointwise observations with independent Gaussian noise,

$$y_i = f(x_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma_y^2), \tag{22}$$

the observations have the marginal distribution

$$\mathbf{y} \mid \boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}_y, \widetilde{\mathbf{K}}(\boldsymbol{\theta}) + \sigma_y^2 \mathbf{I}\right), \tag{23}$$

where

$$[\boldsymbol{\mu}_y]_i = \mu(x_i), \qquad [\widetilde{\mathbf{K}}(\boldsymbol{\theta})]_{ij} \approx k_{\boldsymbol{\theta}}(x_i, x_j). \tag{24}$$

The corresponding marginal log-likelihood is therefore

$$\begin{aligned} \log p(\mathcal{D} \mid \boldsymbol{\theta}) = &-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_y)^\top \left(\widetilde{\mathbf{K}}(\boldsymbol{\theta}) + \sigma_y^2 \mathbf{I}\right)^{-1} (\mathbf{y} - \boldsymbol{\mu}_y) \\ &- \frac{1}{2} \log \det \left(\widetilde{\mathbf{K}}(\boldsymbol{\theta}) + \sigma_y^2 \mathbf{I}\right) - \frac{|X_y|}{2} \log(2\pi). \end{aligned} \tag{25}$$

Consequently, this contribution can be evaluated exactly and differentiated with respect to the kernel hyperparameters without introducing additional MC error.

Looking at Eq. 25, we note that we already have the Cholesky decomposition of the kernel matrix so the log-determinant is easily calculable. This evidence term is important because the conditional term alone often admits degenerate solutions, such as reducing the output variance until a residual is small without explaining the observed data.

Algorithm 1 Weight-space kernel hyperparameter optimisation

Require: Condition $\mathcal{C}$, Gaussian observations $(X_y, \mathbf{y})$, particles B, terminal draws S, update interval q, number of steps K, number of features R

Sample initial flow states $\mathbf{z}_1^{(b)} \sim \mathcal{N}(0, I_R)$, $b = 1, \dots, B$
Draw and fix terminal-sampling noise $\epsilon^{(b,s)}$
for $k = 0, \dots, K-1$ **do**
    Advance $\mathbf{z}_{t_k}^{(1:B)}$ to $\mathbf{z}_{t_{k+1}}^{(1:B)}$ using the guided flow
    if $(k+1) \bmod q = 0$ **then**
        Draw detached terminal weights

$$\boldsymbol{\xi}_0^{(b,s)} \sim p_{\boldsymbol{\theta}}(\boldsymbol{\xi}_0 \mid \mathbf{z}_{t_{k+1}}^{(b)}), \qquad b = 1{:}B,\ s = 1{:}S$$

        using the fixed noise
        Refresh the basis quantities depending on $\boldsymbol{\theta}$
        Compute

$$\widehat{J}(\boldsymbol{\theta}) = \frac{1}{B}\sum_{b=1}^{B} \log\left[\frac{1}{S}\sum_{s=1}^{S} p_\theta(\mathcal{C} \mid \boldsymbol{\xi}_0^{(b,s)})\right] + \log p_{\boldsymbol{\theta}}(\mathbf{y} \mid X_y, \boldsymbol{\theta})$$

        Update $\boldsymbol{\theta} \leftarrow \mathrm{Adam}\left(\boldsymbol{\theta}, -\nabla_{\boldsymbol{\theta}} \widehat{J}(\boldsymbol{\theta})\right)$
        Refresh the basis, decoder, and guidance condition
    end if
end for
Return $\widetilde{f}_{\boldsymbol{\theta}}^{(b)} = \mu + \boldsymbol{\psi}_{\boldsymbol{\theta}}^\top \boldsymbol{\xi}_0^{(b)}$

B.1 Algorithm

A schematic algorithm is given in Algorithm. 1, and described in more detail below.

We use the same frozen standard-normal draws across updates, so stochastic variation in the estimator does not obscure changes caused by $\boldsymbol{\theta}$. Each update first samples S terminal weights from the whitened flow state, before rebuilding all quantities dependent on $\boldsymbol{\theta}$ (including posterior Cholesky factors). Then the objective function and its gradient are computed, using MC samples for the non-conjugate data, and exactly for any conjugate condition. After the optimiser step, quantities depending on $\boldsymbol{\theta}$ are recomputed without retaining the computational graph. In basis space this refreshes the feature matrix and posterior mean, preventing subsequent SDE steps from using a decoder or likelihood built at stale hyperparameters.

C Experimental Details & Additional Results

C.1 Probabilistic downscaling

Experimental details. We compare FLOWGP and our weight-space flow on a two-dimensional non-linear downscaling problem over the unit square. The Gaussian process prior uses a zero mean and a scaled radial basis function kernel. Its initial length scale and output scale are set to $\ell = 0.3$ and $\sigma_f^2 = 1.0$, respectively. Kernel hyperparameters used to generate the ground truth are obtained by maximizing the marginal likelihood of 17 observations. Nine observations with value 5 are placed along the main diagonal, while eight observations with value -5 are placed along the anti-diagonal. The observation-noise variance is 10^{-4}.

A ground-truth function $\hat{f}$ is sampled from a decoupled random Fourier feature representation using 4096 random features. From this function, we noisily query $|X_y| = 32$ uniformly distributed point observations, which define the conjugate Gaussian-process posterior used by both flow parameterisations. The weight-space flow uses a posterior GP approximation (Sec. A.3) containing $R_f = 1024$ random Fourier features, whereas the function-space flow represents the Gaussian process directly at the evaluation points.

The spatial domain is partitioned into an 8×8 array of non-overlapping regions. The observation associated with each region is the maximum value of the ground-truth function within that region. Thus, the non-linear summary operator is

$$\mathcal{A}_j(f) = \max_{x \in \Omega_j} f(x), \qquad j = 1, \ldots, 64, \tag{26}$$

where Ω_j denotes the jth region. Conditional sampling is performed using a Gaussian likelihood,

$$p(\mathcal{C} \mid f) \propto \exp\left\{ -\frac{1}{2\sigma_{\mathcal{C}}^2} \sum_{j=1}^{64} \left[\mathcal{A}_j(f) - c_j\right]^2 \right\}, \tag{27}$$

where $c_j = \mathcal{A}_j(\hat{f})$ and $\sigma_{\mathcal{C}} = 10^{-3}$. In weight space, the field is decoded from its weights before applying the regional maximum operator.

For each configuration, we generate $B = 50$ samples using 1000 SDE integration steps and conditional guidance is estimated using a single Monte Carlo terminal draw per particle.

We evaluate both the initial kernel (length scale $\ell = 0.3$ and output scale $\sigma_f^2 = 1.0$) and an adaptively fitted kernel. During adaptation, the kernel hyperparameters are updated every $q = 10$ integration steps using Adam [22] with learning rate 5×10^{-2}. The conditional objective is estimated using $S = 2$ terminal draws per particle and is combined with the Gaussian-process marginal log-likelihood. After adaptation, the flow is reconstructed using the selected length scale and output scale, and an ordinary guided sampling run is performed with these hyperparameters fixed.

FLOWGP is evaluated on square grids with side lengths

$$H \in \{16, 24, 32, 40, 48, 64\}, \tag{28}$$

corresponding to a maximum of $N = H^2 = 4096$ spatial points. Our weight-space flow is evaluated on grids with side lengths

$$H \in \{16, 24, 32, 48, 64, 88, 128\}, \tag{29}$$

corresponding to a maximum of $N = 16384$ points. The two methods are compared visually at the largest common resolution, $H = 64$. At each resolution, the number of summary regions remains fixed at 64.

Performance is measured using setup time, sampling time, and the domain-averaged root mean squared error

$$\mathrm{RMSE}(f, \hat{f}) = \left[\frac{1}{N} \sum_{i=1}^{N} \big(f(x_i) - \hat{f}(x_i)\big)^2 \right]^{1/2} \tag{30}$$

between the ground-truth function and each sample produced by a given flow. We report the median and interquartile range of the RMSE over the 50 generated samples. All experiments use a fixed random seed.

Adaptive hyperparameter trajectories. In Fig. 3, we display the trajectories of the two kernel hyperparameters ℓ and σ_f^2 over the adaptive sampling run, as described above. The kernel hyperparameters used to generate the ground-truth field are shown with a black dashed line. The non-linear summary operator we use in this experiment only allows first term in the objective function to "see" the maximum value in each summary region, therefore partially missing the minima of the ground-truth field. This offers some explanation as to why the optimiser compresses the output scale of the sampled fields, as compared to the parameters used to generate the ground-truth. By comparison, the lengthscale selected by the optimiser much more closely reflects that of the true field.

Loss curves. In Fig. 4, we show the relative contributions of the two terms in the objective function throughout the adaptive flow. While the hyperparameter trajectories are similar between FLOWGP and the weight-space flow according to Fig. 3, we observe that the non-conjugate loss curve for FLOWGP is much noisier than in weight-space. This may suggest that the optimisation objective is fundamentally more challenging in function space and warrants further investigation.

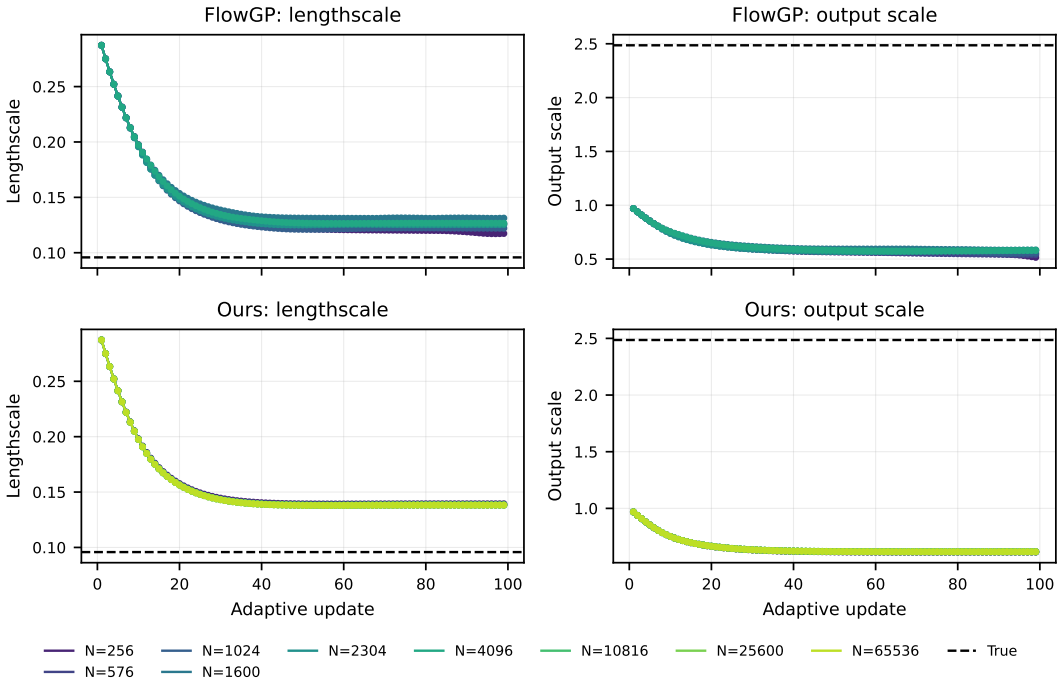

Figure 3: Hyperparameter adaptation histories in both function- (**top row**) and weight-space (**bottom row**) for the downscaling problem. In each case, the lengthscale and kernel output scale used for the ground-truth field is shown with the black dashed line.

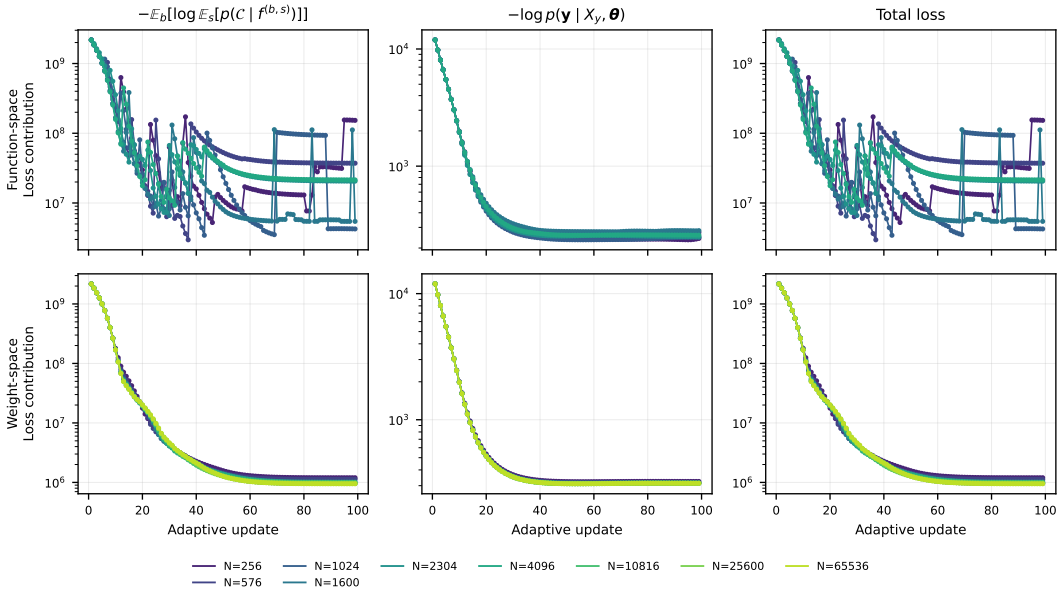

Figure 4: Loss curves for both flow parameterisations (**right**), separated into non-conjugate (**left**) and conjugate (**centre**) contributions. The non-conjugate loss optimises more noisily for FLOWGP than in our weight-space formulation.

C.2 PDE solutions

Experimental details. We compare FLOWGP and our weight-space flow on a two-dimensional stationary Allen-Cahn problem over an irregular-shaped subset of the unit square. The target field satisfies

$$af(x) - bf(x)^3 - c\Delta f(x) = 0, \tag{31}$$

with coefficients $a = b = 1$ and $c = 10^{-3}$, together with a homogeneous Dirichlet condition on the exterior of the domain.

The GP prior has zero mean and a scaled radial basis function kernel. Its length scale and output scale are fixed to $\ell = 0.04$ and $\sigma_f^2 = 2.0$, respectively. The prior is conditioned on $|X_y| = 3$ point observations placed uniformly at random on grid points inside the domain. Each observation has value 1 with observation-noise variance 10^{-4}. The same observations are used to construct the conjugate GP posterior for both flow parameterisations.

FLOWGP represents the GP directly on a 90×90 grid, giving $N = 8100$ evaluation points. The Laplacian is approximated on the strict interior of the entire square grid using the five-point central-difference stencil,

$$(\Delta \mathbf{f})_{i,j} \approx \frac{\mathbf{f}_{i+1,j} - 2\mathbf{f}_{i,j} + \mathbf{f}_{i-1,j}}{\Delta y^2} + \frac{\mathbf{f}_{i,j+1} - 2\mathbf{f}_{i,j} + \mathbf{f}_{i,j-1}}{\Delta x^2}. \tag{32}$$

The associated non-linear condition is therefore defined by

$$\mathcal{A}_{\mathrm{PDE}}(f) = af - bf^3 - c\Delta f, \quad p(\mathcal{C} \mid f) \propto \exp\left\{ -\frac{|\mathcal{A}_{\mathrm{PDE}}(f)|_2^2}{2\sigma_{\mathrm{PDE}}^2} - \frac{|f_{\mathrm{ext}}|_2^2}{2\sigma_{\mathrm{BC}}^2} \right\}, \tag{33}$$

where f_{ext} contains the values outside the domain, $\sigma_{\mathrm{PDE}} = 10^{-6}$, and $\sigma_{\mathrm{BC}} = 10^{-7}$. Choosing the two conditions to be applied equally causes the boundary condition to be violated catastrophically, so a sharper likelihood is applied to the pointwise condition.

The weight-space flow uses a pathwise-conditioned Laplacian-eigenfunction approximation containing $R_f = 256$ spectral features. Because the posterior construction includes one observation-noise feature per observation, its total latent dimension is $R = R_f + |X_y| = 259$. The eigenfunctions are computed according to [20] using a discrete Laplacian on the irregular domain. These basis functions encode the domain boundary, while their analytic Laplacians allow the Allen-Cahn residual to be evaluated directly from the weights:

$$\mathcal{A}_{\mathrm{PDE}}(\boldsymbol{\xi}) = ag_{\boldsymbol{\theta}}(\boldsymbol{\xi}) - bg_{\boldsymbol{\theta}}(\boldsymbol{\xi})^3 - c\Delta g_{\boldsymbol{\theta}}(\boldsymbol{\xi}). \tag{34}$$

Consequently, no separate pointwise boundary likelihood is required in weight space.

For each method, we generate $B = 50$ samples using 1000 SDE integration steps. Conditional guidance is estimated using $S = 1$ Monte Carlo terminal draw per particle for both methods. In both cases, the Euclidean norm of the guidance correction is clipped at 10^2.

For the scaling comparison, both methods are evaluated on square grids with side lengths

$$H \in \{32, 38, 45, 53, 64, 76, 90\}, \tag{35}$$

for a maximum of $N = 8100$ grid points. The kernel, basis width, number of samples, guidance settings, integration scheme, equation coefficients, and domain geometry remain fixed across resolutions. For

$$\mathrm{MSE}_{\mathrm{PDE}} = \frac{1}{B(H-2)^2} \sum_{b=1}^{B} \left| \mathcal{A}_{\mathrm{PDE}}(f^{(b)}) \right|_2^2 \tag{36}$$

and the mean-squared exterior boundary residual

$$\mathrm{MSE}_{\mathrm{BC}} = \frac{1}{B|\Omega_h^{\mathrm{c}}|} \sum_{b=1}^{B} \sum_{x_i \in \Omega_h^{\mathrm{c}}} \left[f^{(b)}(x_i) \right]^2, \tag{37}$$

where Ω_h^{c} denotes the grid points outside the irregular domain. To ensure a fair comparison, functional samples from both methods are evaluated using the condition applied to FLOWGP, such that any discrepancy that remains disadvantages our method. All experiments use a fixed random seed.