

Spatial Prediction with Guided Diffusion

Andrew Zammit-Mangion^{1*} Lachlan Astfalck¹ Raphaël Huser²
 Christopher Nemeth³ Henry B. Moss³

¹School of Mathematics and Statistics, UNSW Sydney, Sydney, Australia

²Statistics Program, King Abdullah University of Science and Technology (KAUST),
 Thuwal, Saudi Arabia

³School of Mathematical Sciences, Lancaster University, Lancaster, United Kingdom

Abstract

Spatial prediction is a critically important task in data analyses that involve spatially or spatio-temporally-referenced data. It is commonly carried out using hierarchical models, which comprise a process model describing the latent process of interest, and a data model describing how the process is observed. The target quantity is the predictive distribution, which is the conditional distribution of the latent spatial process given the observed data. Unfortunately, the predictive distribution is rarely available in closed form and can be difficult to sample from. Here, we show how recent developments in guided diffusion modelling from machine learning can address a wide class of problems in spatial prediction when the likelihood function derived from the data model is known and differentiable, possibly after relaxation. To show the versatility of the approach, we apply it to both Gaussian and non-Gaussian process models and a variety of data models that describe direct partial observations, linear and areal aggregates, counts, right censoring, and area-exceedance functionals. Where possible, we validate the framework against exact and Markov-chain-Monte-Carlo (MCMC)-derived predictive distributions. We demonstrate our approach on several simulation studies and in a controlled experiment of methane-flux inversion over the UK and Ireland. We show that guided diffusion can sample from the correct predictive distribution in a variety of settings, that it can be applied to models where MCMC is intractable, and that it scales well with the dimensionality of the latent process.

Keywords: Conditional simulation, Diffusion models, Generative modelling, Non-Gaussian models, Spatial processes

*Corresponding author: a.zammit_mangion@unsw.edu.au

1 Introduction

A key task in spatial data analysis is to predict a spatial process from various datasets or conditioning information. In studies of biodiversity, for example, a common objective is to infer spatial species dissimilarity from presence/absence records (White et al., 2024); in flux inversion studies, the goal is to predict flux (sources and sinks) of a trace gas such as carbon dioxide from both satellite retrievals and in-situ data (Zammit-Mangion et al., 2022; Bertolacci et al., 2024); in air pollution studies, a common objective is to produce a spatial map of exceedance probabilities (Bolin and Lindgren, 2015); and in ice sheet mass balance studies, one needs to predict the volume and mass of ice from satellite measurements (Zammit-Mangion et al., 2015; Martín-Español et al., 2016). These spatial predictions ultimately inform national and international policy and decision making (e.g., Sweet et al., 2017; Byrne et al., 2023).

Spatial prediction can be summarised as follows. Consider a process $\{Y_0(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$ on some domain $\mathcal{D} \subset \mathbb{R}^d, d \geq 1$ modelled using a *process model* $p(Y_0(\cdot))$, and the availability of information that we can condition on for making inferential statements on $Y_0(\cdot)$. This information is most often in the form of data $\mathbf{Z}$ that are related to the process via a *data model* $p(\mathbf{Z} | Y_0(\cdot))$. The target quantity is the predictive distribution given by

$$p(Y_0(\cdot) | \mathbf{Z}) \propto p(Y_0(\cdot), \mathbf{Z}) = \underbrace{p(Y_0(\cdot))}_{\text{process model}} \underbrace{p(\mathbf{Z} | Y_0(\cdot))}_{\text{data model}}. \quad (1)$$

If $\mathbf{Z} | Y_0(\cdot)$ is linear in $Y_0(\cdot)$ and Gaussian, and $Y_0(\cdot)$ is also Gaussian, then the predictive distribution $p(Y_0(\mathbf{s}) | \mathbf{Z}), \mathbf{s} \in \mathcal{D}$, is Gaussian, and typically straightforward to sample from. When the process model $Y_0(\cdot)$ is non-Gaussian, direct evaluation of $p(Y_0(\mathbf{s}) | \mathbf{Z}), \mathbf{s} \in \mathcal{D}$, is not possible, and this predictive distribution is often difficult to sample from. The problem is compounded when the data model is also non-Gaussian and/or non-linear in $Y_0(\cdot)$.

There are several existing approaches for spatial prediction that are tailored to various model choices, each with their own strengths and limitations. A widely applicable approach is Markov chain Monte Carlo (MCMC) which, in many cases, can be used to sample from the target predictive density when this can be evaluated up to a constant of proportionality. Integrated nested Laplace approximations (INLA; Rue et al., 2009) is designed for latent Gaussian models. Approximate Bayesian computation (ABC; Sisson et al., 2018) precludes the need of pointwise evaluation of process densities or the likelihood function, but scales poorly with the dimension of the summary statistics used in the computation. For specific non-Gaussian processes, bespoke conditional simulators have been developed. In the spatial extremes context, for example, Dombry et al. (2013) develop an MCMC algorithm for conditional simulation of Brown–Resnick max-stable processes. These methods are useful in their specific use cases, but each is tied to a specific process family and data model.

More recently, neural conditional simulation (NCS; Walchessen et al., 2025) has been developed to enable fast conditional simulation via score-based diffusion (Song et al., 2021). NCS learns a map from data to the target predictive distribution by training a neural network to directly approximate a score function that is required for conditional simulation. Walchessen et al. (2025) successfully applied their method to the Brown–Resnick spatial process with hundreds of observations, for which MCMC is computationally prohibitive.

However, NCS has a practical limitation: the data model must be specified before training the neural network, and cannot be changed at prediction time. If the data model changes or is incorrectly specified, the neural network has to be re-trained. When the data model is unknown until data are observed, or when several data models are likely to be encountered in a given application, this approach is not practical.

The limitation of NCS can be overcome by leveraging *guided diffusion*, where one first constructs a diffusion model to generate unconditionally (rather than conditionally) from $Y_0(\cdot)$, typically using a neural network that estimates a score function. Then, at prediction time, one combines the unconditional score function and a likelihood gradient to sample from the predictive distribution. Guided diffusion thus keeps the two factors in (1) apart, and only learns the process model in an initial step. Variants of guided diffusion include diffusion posterior sampling (DPS; Chung et al. 2023) and IIGDM (Song et al., 2023). The flexibility of guided diffusion addresses the misspecification risk inherent in NCS, where the neural network is trained using conditioning data that come from a specific data model or family of data models. Guided diffusion has been explored outside spatial statistics in compressed-sensing imaging, inverse problems, and data assimilation (Kawar et al., 2022; Rozet and Louppe, 2023). Our aim is to adapt and validate its use for the task of spatial prediction.

In this paper we show how guided diffusion can be leveraged as a general approach to making prediction of complex processes from heterogeneous data sources. We demonstrate the utility on both the vanilla Gaussian process and on the amplitude field of the complex-valued nonlinear Ginzburg–Landau equation. For data models, we consider cases where information is in the form of direct partial observations, Poisson counts, right-censored measurements, areal aggregates, and area-exceedance functionals. We also demonstrate our framework in a realistic controlled experiment where we infer methane flux from noisy in-situ data. In most cases we compare the predictive distribution to that from MCMC, when it is computationally feasible to do so. We find guided diffusion to be a general and scalable approach to spatial prediction: a single trained process model serves a wide range of data models, and the process model itself need only be available through simulation.

The remainder of the paper is organised as follows. Section 2 frames score-based diffusion in the context of spatial prediction, and outlines NCS. Section 3 presents guided diffusion for spatial prediction. Section 4 presents results from experiments on Gaussian processes and Ginzburg–Landau fields. Section 5 applies the framework in the context of an observing system simulation experiment (OSSE) related to flux inversion. Section 6 concludes.

2 Score-Based Diffusion for Spatial Prediction

Throughout this paper we will consider a discretisation of the process $Y_0(\cdot)$ on a gridding $\mathcal{G}$ of $\mathcal{D}$ comprising n cells, which we denote as $\mathbf{Y}_0 \equiv (Y_0(\mathbf{s}) : \mathbf{s} \in \mathcal{G})^\top$. Consider now the forward diffusion process

$$d\mathbf{Y}_t = -\frac{1}{2}\beta_t\mathbf{Y}_t dt + \sqrt{\beta_t}d\mathbf{W}_t, \quad t \in [0, T], \quad (2)$$

where $\mathbf{W}_t$ is a standard Brownian motion in $\mathbb{R}^n$, $\beta_t > 0$ is a deterministic noise schedule, t denotes diffusion time (which is unrelated to physical time), and where $T > 0$ (we fix

$T = 1$ throughout). It is straightforward to show (Song et al., 2021) that the conditional distribution of $\mathbf{Y}_t$ given $\mathbf{Y}_0$ is

$$\mathbf{Y}_t \mid \mathbf{Y}_0 \sim \text{Gau}(\alpha_t \mathbf{Y}_0, \sigma_t^2 \mathbf{I}_n), \quad (3)$$

where $\text{Gau}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\alpha_t = \exp\left(-\frac{1}{2} \int_0^t \beta_s ds\right)$ and $\sigma_t^2 = 1 - \alpha_t^2$ (see Appendix A.1 for a derivation). At diffusion time $t = T$ the noise schedule is chosen so that $\alpha_T \approx 0$ and $\mathbf{Y}_T \sim \text{Gau}(\mathbf{0}, \mathbf{I}_n)$, regardless of the initial $\mathbf{Y}_0$. Sampling from $p(\mathbf{Y}_0)$ is then equivalent to integrating the corresponding reverse-time stochastic differential equation (Anderson, 1982) from $\mathbf{Y}_T \sim \text{Gau}(\mathbf{0}, \mathbf{I}_n)$ backwards to $\mathbf{Y}_0$:

$$d\mathbf{Y}_t = \left[-\frac{1}{2}\beta_t \mathbf{Y}_t - \beta_t \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)\right] dt + \sqrt{\beta_t} d\bar{\mathbf{W}}_t, \quad (4)$$

where $\bar{\mathbf{W}}_t$ is a Brownian motion in reverse time (see Appendix A.2 for a derivation). The only unknown quantity in (4) is the marginal score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$, which depends on the marginal density of the noised process.

If $\mathbf{Y}_0$ is Gaussian, then the score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$ is available in closed form. Specifically, suppose $\mathbf{Y}_0 \sim \text{Gau}(\mathbf{0}, \mathbf{K})$ for an $n \times n$ prior covariance matrix $\mathbf{K}$ induced by evaluating a spatial covariance function over the grid $\mathcal{G}$. Marginalising (3) over $\mathbf{Y}_0$ we see that $\mathbf{Y}_t \sim \text{Gau}(\mathbf{0}, \alpha_t^2 \mathbf{K} + \sigma_t^2 \mathbf{I}_n)$, which leads to the marginal score

$$\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t) = -(\alpha_t^2 \mathbf{K} + \sigma_t^2 \mathbf{I}_n)^{-1} \mathbf{Y}_t. \quad (5)$$

Substituting (5) into the reverse-time SDE (4) yields a linear time-varying system that can be numerically integrated, with no neural network required (Moss et al., 2026). We use this property in the Gaussian-process experiment of Section 4.1.

For a general non-Gaussian process, the marginal $p(\mathbf{Y}_t)$, and therefore its score, is intractable, and one must instead estimate it. The usual strategy is to use a neural network for score estimation, which we denote by $\mathbf{s}_\phi(\mathbf{Y}_t, t) : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}^n$, where ϕ are neural network parameters. The neural network can be trained to be an estimator of the score by minimising the score-matching objective (Vincent, 2011; Song et al., 2021):

$$\mathcal{L}_S(\phi) = \frac{1}{T} \int_0^T \mathbb{E}_{\mathbf{Y}_0} \mathbb{E}_{\mathbf{Y}_t \mid \mathbf{Y}_0} \left[\|\mathbf{s}_\phi(\mathbf{Y}_t, t) - \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t \mid \mathbf{Y}_0)\|^2 \right] dt, \quad (6)$$

where $\mathbf{Y}_0 \sim p(\mathbf{Y}_0)$ and $\mathbf{Y}_t \mid \mathbf{Y}_0$ follows from (3); see Appendix A.3 for a derivation. The quantity $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t \mid \mathbf{Y}_0) = -(\mathbf{Y}_t - \alpha_t \mathbf{Y}_0)/\sigma_t^2$ is numerically unstable for small σ_t (that occurs when $t \rightarrow 0$). Therefore, we adopt the approach of training a neural network to estimate the scaled score $-\sigma_t \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$ instead (Ho et al., 2020). Let $\sigma_t \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t \mid \mathbf{Y}_0) \equiv -\mathbf{r}_t$, where the scaled residual $\mathbf{r}_t \equiv (\mathbf{Y}_t - \alpha_t \mathbf{Y}_0)/\sigma_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. We denote the approximating neural network by $\mathbf{r}_\phi(\mathbf{Y}_t, t) : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}^n$, where ϕ are neural network parameters. The loss function we minimise is then

$$\mathcal{L}_R(\phi) = \frac{1}{T} \int_0^T \mathbb{E}_{\mathbf{Y}_0} \mathbb{E}_{\mathbf{r}_t} \left[\|\mathbf{r}_\phi(\alpha_t \mathbf{Y}_0 + \sigma_t \mathbf{r}_t, t) - \mathbf{r}_t\|^2 \right] dt. \quad (7)$$

Once trained, using the same arguments in Appendix A.3, $\mathbf{r}_\phi(\mathbf{Y}_t, t) \approx -\sigma_t \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$ for $t \in [0, T]$, and therefore $-\mathbf{r}_\phi(\mathbf{Y}_t, t)/\sigma_t \approx \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$. In practice (7) is minimised using

Monte Carlo approximation and stochastic gradient descent. Unconditional samples from $p(\mathbf{Y}_0)$ are obtained by integrating (4) backward in time, with $-\mathbf{r}_\phi(\cdot, \cdot)/\sigma_t$ in place of the true score; see Section S.1 of the Supplementary Material for details. We adopt a neural network to characterise $p(\mathbf{Y}_0)$ in this way in the Ginzburg–Landau experiment of Section 4.2 and the methane flux inversion experiment of Section 5.

Neural conditional simulation So far we have discussed how we could use diffusion models to generate *unconditional* samples from $p(\mathbf{Y}_0)$. As we show in Section 3, these samples can be progressively “guided” to predictive samples in accordance with different data models. This is in stark contrast to NCS where one instead trains a network to directly generate *conditional* samples (i.e., from $p(\mathbf{Y}_0 | \mathbf{Z})$), through a neural network that targets the conditional score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t | \mathbf{Z})$ (Walchessen et al., 2025). NCS is typically implemented using a neural network $\tilde{\mathbf{r}}_\phi(\cdot, \cdot, \cdot)$ that also takes $\mathbf{Z}$ as input, and that is trained using the loss

$$\mathcal{L}_{\text{NCS}}(\phi) = \frac{1}{T} \int_0^T \mathbb{E}_{\mathbf{Y}_0} \mathbb{E}_{\mathbf{Z}|\mathbf{Y}_0} \mathbb{E}_{\mathbf{r}_t} [\|\tilde{\mathbf{r}}_\phi(\alpha_t \mathbf{Y}_0 + \sigma_t \mathbf{r}_t, \mathbf{Z}, t) - \mathbf{r}_t\|^2] dt. \quad (8)$$

It is a straightforward extension of the derivation in Appendix A.3 to show that minimising (8) makes $-\tilde{\mathbf{r}}_\phi(\mathbf{Y}_t, \mathbf{Z}, t)/\sigma_t$ target the required conditional score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t | \mathbf{Z})$ since $\mathbf{Z} \perp \mathbf{Y}_t | \mathbf{Y}_0$. For prediction, NCS draws predictive samples by running the reverse SDE with the conditional score in place of the unconditional one. There is one notable drawback of NCS: If, at prediction time, the data do not follow the data model $p(\mathbf{Z} | \mathbf{Y}_0)$ used to train the network, then samples from NCS do not target the correct predictive distribution. Retraining for every possible data model can be infeasible in several applications. This is the limitation that the guided diffusion approach, presented in Section 3, removes.

3 Guided Diffusion for Spatial Prediction with Various Data Models

3.1 Score decomposition and approximate guidance

The reverse-time diffusion (4) samples from whichever distribution the score belongs to. For example, when the score is $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$, reverse diffusion returns draws from the process model $p(\mathbf{Y}_0)$. To sample from the predictive distribution $p(\mathbf{Y}_0 | \mathbf{Z})$ we therefore need to replace the unconditional score in (4) with the conditional score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t | \mathbf{Z})$. A simple application of Bayes’ rule then reveals a decomposition of the score into that of the process model and that of the data model:

$$\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t | \mathbf{Z}) = \underbrace{\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)}_{\text{process model score}} + \underbrace{\nabla_{\mathbf{Y}_t} \log p(\mathbf{Z} | \mathbf{Y}_t)}_{\text{data model score}}. \quad (9)$$

Equation (9) reveals a way forward to defer the choice of data model when training a neural network to estimate the score function. In particular, $\mathbf{r}_\phi(\cdot, \cdot)$ trained using (7) to estimate the process model score can be re-used with a wide range of data models at prediction

time (Lai et al., 2025). Importantly, no additional neural network is needed to estimate the data-model score, which in several cases can be directly evaluated.

While elegant, a drawback of this approach is that, although we have access to $p(\mathbf{Z} \mid \mathbf{Y}_0)$, we do not usually have access to $p(\mathbf{Z} \mid \mathbf{Y}_t)$ since,

$$p(\mathbf{Z} \mid \mathbf{Y}_t) = \int p(\mathbf{Z} \mid \mathbf{Y}_0) p(\mathbf{Y}_0 \mid \mathbf{Y}_t) d\mathbf{Y}_0 \quad (10)$$

is only available in closed form in rare cases, such as when both $p(\mathbf{Z} \mid \mathbf{Y}_0)$ and $p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ are Gaussian. There are several approaches to circumvent this issue, developed largely in the image-generation literature: diffusion posterior sampling (DPS; Chung et al., 2023) collapses $p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ to a point mass at its mean; importance sampling and its sequential Monte Carlo variants approximate the integral directly; and pseudoinverse-guided diffusion (PIGDM; Song et al., 2023) replaces $p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ with a Gaussian surrogate. We describe each in turn.

Tweedie’s identity (Efron, 2011) gives an expression for the conditional expectation of $\mathbf{Y}_0$ given $\mathbf{Y}_t$,

$$\hat{\mathbf{Y}}_0(\mathbf{Y}_t) \equiv \mathbb{E}[\mathbf{Y}_0 \mid \mathbf{Y}_t] = \frac{1}{\alpha_t} (\mathbf{Y}_t + \sigma_t^2 \nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)) \approx \frac{1}{\alpha_t} (\mathbf{Y}_t - \sigma_t \mathbf{r}_\phi(\mathbf{Y}_t, t)), \quad (11)$$

where the approximation uses the estimated score (via the neural network) in place of the true score. The map $\mathbf{Y}_t \mapsto \hat{\mathbf{Y}}_0(\mathbf{Y}_t)$ can be computed at each diffusion step without any additional forward pass. DPS then approximates (10) by collapsing the integral (10) to the Tweedie estimate,

$$p(\mathbf{Z} \mid \mathbf{Y}_t) \approx p(\mathbf{Z} \mid \hat{\mathbf{Y}}_0(\mathbf{Y}_t)), \quad (12)$$

neglecting the uncertainty in $\mathbf{Y}_0 \mid \mathbf{Y}_t$, so that the data model score becomes

$$\nabla_{\mathbf{Y}_t} \log p(\mathbf{Z} \mid \mathbf{Y}_t) \approx \nabla_{\mathbf{Y}_t} \log p(\mathbf{Z} \mid \hat{\mathbf{Y}}_0(\mathbf{Y}_t)), \quad (13)$$

where the right-hand side is in practice evaluated using automatic differentiation.

Equation (12) ignores the conditional variance $\text{Cov}(\mathbf{Y}_0 \mid \mathbf{Y}_t)$, which leads DPS to generate conditional samples that are often under-dispersed relative to the true predictive distribution. Alternatively, one could consider instead drawing $\mathbf{Y}_0^{(s)} \sim p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ and estimating (10) and its score through importance sampling, but the importance weights tend to be degenerate in even moderately-high dimensional settings. There are sequential-Monte-Carlo variants designed to alleviate this, but this comes at the high computing cost of running a particle filter (Wu et al., 2023; Cardoso et al., 2024).

Finally, one could instead adopt IIGDM, which relaxes the point mass assumption of DPS but that also avoids the degeneracy of importance sampling. Here, one approximates $p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ using a Gaussian surrogate,

$$p(\mathbf{Y}_0 \mid \mathbf{Y}_t) \approx \text{Gau}(\hat{\mathbf{Y}}_0(\mathbf{Y}_t), \mathbf{V}_t), \quad \mathbf{V}_t = \left(\mathbf{K}^{-1} + \frac{\alpha_t^2}{\sigma_t^2} \mathbf{I}_n \right)^{-1}, \quad (14)$$

where recall that $\mathbf{K} \equiv \text{Cov}(\mathbf{Y}_0)$. The expression for $\mathbf{V}_t$ can be derived using the identity $\text{Cov}(\mathbf{Y}_0 \mid \mathbf{Y}_t) = \text{Cov}(\mathbf{Y}_0) - \text{Cov}(\mathbf{Y}_0, \mathbf{Y}_t) \text{Cov}(\mathbf{Y}_t)^{-1} \text{Cov}(\mathbf{Y}_t, \mathbf{Y}_0)$ and the Sherman–Morrison–Woodbury identity. When $\mathbf{Y}_0$ is Gaussian, this approximation is exact since, in that case,

$\mathbf{Y}_0 \mid \mathbf{Y}_t$ is Gaussian with mean $\hat{\mathbf{Y}}_0(\mathbf{Y}_t)$ and covariance $\mathbf{V}_t$. In the Gaussian setting of Section 4.1 we follow Moss et al. (2026) and run the diffusion in whitened coordinates $\mathbf{Y}_0^W \equiv \mathbf{K}^{-1/2}(\mathbf{Y}_0 - \mathbb{E}[\mathbf{Y}_0])$; applying (14) in those coordinates gives $\mathbf{V}_t = \sigma_t^2 \mathbf{I}_n$ (since, as seen in Appendix A.1, $\sigma_t^2 + \alpha_t^2 = 1$), which remains exact. When $\text{Cov}(\mathbf{Y}_0)$ is not known, we approximate it with $\varsigma^2 \mathbf{I}_n$, where ς^2 is the sample variance of the data used for training the neural network. In practice $\varsigma^2 = 1$ since the data are typically standardised before training the network, in which case, once again $\mathbf{V}_t = \sigma_t^2 \mathbf{I}_n$.

Plugging the Gaussian approximation to $p(\mathbf{Y}_0 \mid \mathbf{Y}_t)$ into (10), we obtain

$$p(\mathbf{Z} \mid \mathbf{Y}_t) \approx \int p(\mathbf{Z} \mid \mathbf{Y}_0) \text{Gau}(\mathbf{Y}_0; \hat{\mathbf{Y}}_0(\mathbf{Y}_t), \mathbf{V}_t) d\mathbf{Y}_0, \quad (15)$$

where $\text{Gau}(\mathbf{Y}_0; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\mathbf{Y}_0$. This quantity, often referred to as the ‘noised likelihood’, is available in closed form with linear-Gaussian data models. In the general non-Gaussian case, a further approximation is needed to evaluate (15). We detail the approximations we use when discussing the data models in Sections 3.2 and 3.3. Because of the extra approximations required, implementing IIGDM can be more laborious than implementing DPS. However, as we will see in the experiments of Section 4, its use generally leads to predictive intervals with empirical coverage closer to the nominal level.

3.2 Guided diffusion with linear-Gaussian data models

A large fraction of spatial data models used in practice share the linear-Gaussian form

$$\mathbf{Z} = \mathbf{H}\mathbf{Y}_0 + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \text{Gau}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}_m), \quad (16)$$

where $\mathbf{Z} \in \mathbb{R}^m$ are noisy data, $\mathbf{H}$ is an $m \times n$ matrix that is (typically) known, $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is measurement error, and σ_ϵ^2 is measurement-error variance. This data model includes the ubiquitous case of direct (partial) observations, where $\mathbf{H}$ is an incidence matrix. It also includes the important case of linear spatial aggregates, where $\mathbf{H}$ averages the field over disjoint regions:

$$(\mathbf{H}\mathbf{Y}_0)_k = |\mathcal{R}_k|^{-1} \sum_{i \in \mathcal{R}_k} Y_0(\mathbf{s}_i), \quad k = 1, \dots, m,$$

for a partitioning $\{\mathcal{R}_k\}_{k=1}^m$ of the underlying grid into m regions. Another important special case of (16) is a soft mass-conservation constraint, where $\mathbf{H} \equiv \mathbf{h}^\top = \frac{1}{n} \mathbf{1}^\top$ is a $1 \times n$ incidence vector, and the now scalar observation Z is set to some constraint value with σ_ϵ^2 ‘small’. In such cases, the Gaussian log-likelihood evaluated at Tweedie’s estimate is, for known σ_ϵ^2 ,

$$\log p(\mathbf{Z} \mid \hat{\mathbf{Y}}_0(\mathbf{Y}_t)) = -\frac{1}{2\sigma_\epsilon^2} (\mathbf{Z} - \mathbf{H}\hat{\mathbf{Y}}_0(\mathbf{Y}_t))^\top (\mathbf{Z} - \mathbf{H}\hat{\mathbf{Y}}_0(\mathbf{Y}_t)) + \text{const},$$

and the gradient required for DPS in (13) is given by

$$\nabla_{\mathbf{Y}_t} \log p(\mathbf{Z} \mid \hat{\mathbf{Y}}_0(\mathbf{Y}_t)) = \frac{1}{\sigma_\epsilon^2} \left(\frac{\partial \hat{\mathbf{Y}}_0(\mathbf{Y}_t)}{\partial \mathbf{Y}_t} \right)^\top \mathbf{H}^\top (\mathbf{Z} - \mathbf{H}\hat{\mathbf{Y}}_0(\mathbf{Y}_t)). \quad (17)$$

The Jacobian $\partial \hat{\mathbf{Y}}_0(\mathbf{Y}_t)/\partial \mathbf{Y}_t = \alpha_t^{-1}(\mathbf{I}_n - \sigma_t \partial \mathbf{r}_\phi(\mathbf{Y}_t, t)/\partial \mathbf{Y}_t)$ follows from differentiating (11). It involves the derivative of $\mathbf{r}_\phi(\mathbf{Y}_t, t)$ with respect to $\mathbf{Y}_t$, which in practice is evaluated via automatic differentiation. For this data model no further approximation is needed when evaluating the noised likelihood (15) for Π GDM, since the integral therein is available in closed form.

3.3 Guided diffusion with non-linear and non-Gaussian data models

It is often the case that spatial data are related to the underlying process in a non-Gaussian or nonlinear way. One of the most common non-Gaussian data models in the spatial setting is a count data model, specifically a conditional Poisson model of the form

$$\mathbf{Z} \mid \mathbf{Y}_0 \stackrel{\text{ind}}{\sim} \text{Poisson}(\rho \tilde{\mathbf{Y}}_0), \quad (18)$$

where $\tilde{\mathbf{Y}}_0 \equiv \mathbf{H}g(\mathbf{Y}_0)$, $g: \mathbb{R} \rightarrow \mathbb{R}_+$ is a differentiable function applied elementwise, $\rho > 0$ is a global intensity parameter, and where the count data $Z_1, \dots, Z_m$ are conditionally independent. The log-Gaussian Cox process is often represented in this way (e.g., Cseke et al., 2016), but note that our framework does not require a Gaussian latent field. As in the Gaussian case, the Jacobian can easily be computed and incorporated into guided samplers. In Section 4.1 we let $\mathbf{H}$ be an $m \times n$ incidence matrix identifying which pixels are observed. Details on how we approximate (15) for Π GDM with this data model are given in Appendix B.1.

Another data model commonly encountered with spatial data is a right-censored observation model. For a known threshold c , we observe a noisy value $Z_i = Y_{0,i} + \epsilon_i$ with $\epsilon_i \sim \text{Gau}(0, \sigma_\epsilon^2)$ only when this measurement does not exceed c ; if $Z_i > c$ the cell is censored in the upper tail and we only know that the measurement lies above c . Let $\mathbf{M} \equiv (M_1, \dots, M_n)^\top$, where $M_i = 1$ corresponds to a censored (i.e., masked) data point. Censoring at c means $\Pr(Z_i > c \mid Y_{0,i}) = \Phi((Y_{0,i} - c)/\sigma_\epsilon)$, so

$$\log p(\mathbf{Z}, \mathbf{M} \mid \mathbf{Y}_0) = \sum_{i: M_i=0} \left[-\frac{1}{2} \left(\frac{Z_i - Y_{0,i}}{\sigma_\epsilon} \right)^2 \right] + \sum_{i: M_i=1} \log \Phi \left(\frac{Y_{0,i} - c}{\sigma_\epsilon} \right), \quad (19)$$

up to an additive constant independent of $\mathbf{Y}_0$. This censored log-likelihood is smooth in $Y_{0,i}$ and differentiable. Details on how we approximate (15) for Π GDM with this data model are given in Appendix B.2.

A related data model we consider is an ‘area-exceedance’ data model, where a single scalar observation $Z = \sum_{i=1}^n \mathbf{1}\{Y_{0,i} > c\}$ counts the number of grid cells in which the field exceeds a known threshold c . Such a summary finds use in ‘what-if’ experiments, where one is interested in the distribution of the process conditional on a certain proportion of the domain experiencing extreme events. In this case, we refer to Z as an area-exceedance functional; it is a non-linear scalar functional of the entire field and distinct from the linear spatial aggregates of Section 3.2. We can make the likelihood differentiable, as required by guided diffusion, by letting $\sigma_\kappa(Y_{0,i}) \equiv \text{expit}(\kappa(Y_{0,i} - c))$, $\kappa > 0$, and endowing Z with a low-variance discrete distribution on $\{0, \dots, n\}$, centred at $h(\mathbf{Y}_0) \equiv \sum_{i=1}^n \sigma_\kappa(Y_{0,i})$. Here we

use a discrete version of the Gaussian distribution:

$$\log \Pr(Z = k \mid \mathbf{Y}_0; \kappa, \xi) = -\frac{(k - h(\mathbf{Y}_0))^2}{2\xi^2} - \log \sum_{j=0}^n \exp\left(-\frac{(j - h(\mathbf{Y}_0))^2}{2\xi^2}\right), \quad k \in \{0, \dots, n\}. \quad (20)$$

The variance ξ^2 can be set small; in Section 4.1 we take $\xi^2 = 0.1$, which concentrates almost all mass within one count of $h(\mathbf{Y}_0)$. Details on how we approximate (15) for IIGDM with this data model are given in Appendix B.3.

4 Experiments

In this section we consider both Gaussian (Section 4.1) and non-Gaussian (Section 4.2) spatial process models, together with linear-Gaussian and non-Gaussian data models. On the process side, the Gaussian process model of Section 4.1 has a score function that is available in closed form, while the non-Gaussian model of Section 4.2 has an intractable score function that we replace with a trained neural network. On the data side, the linear-Gaussian model (16) gives a noised likelihood (10) that is also available in closed form, while the non-Gaussian data models of Section 3.3 require the approximations discussed in Section 3.1. Details on how the diffusion model was implemented for these experiments and that in Section 5 are given in Section S.1 of the Supplementary Material. In Section S.2 of the Supplementary Material we also compare guided diffusion to NCS (Walchessen et al., 2025). There, we show that the two approaches are competitive when the data model matches the one used at training time, but that only guided diffusion accommodates a subsequent change in the data model. We use a 32×32 regular grid ($n = 1024$) to discretise $\mathcal{D} = [0, 1]^2$ in the Gaussian process experiment of Section 4.1, and $\mathcal{D} = [0, 40]^2$ in the Ginzburg–Landau experiment of Section 4.2. We let $\mathbf{Y}_0$ be $Y_0(\cdot)$ evaluated on this grid.

4.1 Gaussian process model

Let $Y_0(\cdot)$ be a mean-zero Gaussian process with a Matérn covariance function of smoothness $\nu = 3/2$, given by

$$\text{Cov}(Y_0(\mathbf{s}_i), Y_0(\mathbf{s}_j)) = \sigma_Y^2 \left(1 + \frac{\sqrt{3} \|\mathbf{s}_i - \mathbf{s}_j\|}{\ell}\right) \exp\left(-\frac{\sqrt{3} \|\mathbf{s}_i - \mathbf{s}_j\|}{\ell}\right).$$

We set the marginal variance $\sigma_Y^2 = 1$, and length scale $\ell = 0.2$. Recall from Section 2 that, because the gridded $\mathbf{Y}_0$ is Gaussian, the marginal score $\nabla_{\mathbf{Y}_t} \log p(\mathbf{Y}_t)$ is linear in $\mathbf{Y}_t$ and available in closed form. Throughout these Gaussian-process experiments we diffuse the whitened field, as described below (14) and in more detail in Moss et al. (2026), so that $\mathbf{V}_t = \sigma_t^2 \mathbf{I}_n$ in the whitened coordinates and $\sigma_t^2 \mathbf{K}$ in the original ones.

Point-referenced Gaussian observations We first consider the data model (16). We simulate 100 realisations from the underlying spatial process and, from each, we generate a dataset by constructing $\mathbf{H}$ so that a random 10% of the grid cells are observed ($m = 102$)

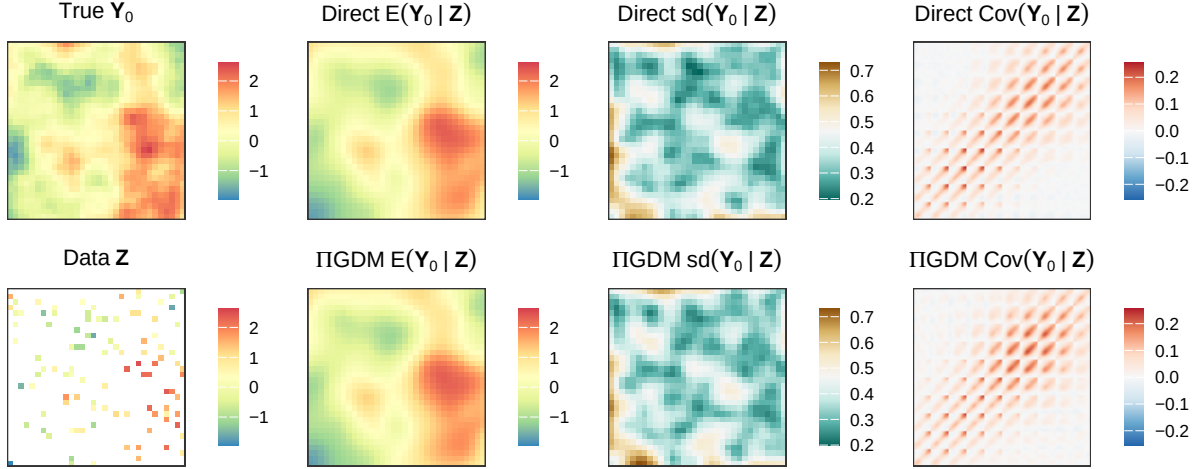


Figure 1: Predictive distributions from synthetic Gaussian data generated from a Matérn 3/2 Gaussian process. Left: true field $\mathbf{Y}_0$ (top) and partial observations $\mathbf{Z}$ (bottom). Centre-left: predictive mean from direct sampling (top) and IIGDM (bottom). Centre-right: predictive standard deviation from direct sampling (top) and IIGDM (bottom). Right: predictive covariance of the central 10×10 block of grid cells from direct sampling (top) and IIGDM (bottom).

with additive Gaussian measurement-error variance $\sigma_\epsilon^2 = 0.2$. For each realisation we draw 200 predictive samples in three ways: directly from the closed-form Gaussian predictive distribution (Rasmussen and Williams, 2006, Chapter 2), and via guided reverse-time diffusion with 500 discretization steps using both DPS and IIGDM guidance. We use these samples to compute the root-mean-square error (RMSE) between the predictive mean and the true field over all $n = 1024$ grid cells, the empirical coverage of nominal 90% pointwise prediction intervals constructed from equal-tailed empirical quantiles, the field-level energy score (ES), and the averaged per-cell continuous ranked probability score (CRPS) (see Gneiting and Raftery, 2007, for details).

In Figure 1 we show results from the first realisation when using IIGDM. We see that both the marginal and the joint properties of the predictive distribution are accurately captured through guidance. Table 1 summarises the diagnostics averaged over the 100 realisations. There is some loss in fidelity when using DPS over direct sampling from the predictive Gaussian distribution, as expected. On the other hand, guidance using IIGDM is exact up to time discretisation in this case.

Poisson observations We now consider the Poisson data model (18) with $g(\cdot) \equiv \exp(\cdot)$, for which the predictive distribution has no closed form. We take the global intensity $\rho = 1$. As in the Gaussian case, we construct $\mathbf{H}$ so that a random 10% of the grid cells are observed, and validate against predictions from an elliptical slice sampler (ESS; Murray et al., 2010). We simulate 100 realisations from the underlying spatial process and, from each, generate a synthetic count dataset. For each realisation we draw 200 predictive samples in three

Table 1: Diagnostics for predictive samples generated directly from the Gaussian predictive distribution (Direct), via DPS guidance, and via IIGDM guidance, for the Gaussian-process experiment of Section 4.1. Entries are means over 100 realisations, with standard deviation shown in parentheses.

Sampler	RMSE	90% coverage	ES	CRPS
Direct	0.389 (0.040)	0.89 (0.03)	8.78 (0.90)	0.214 (0.020)
DPS	0.398 (0.040)	0.85 (0.04)	9.02 (0.95)	0.220 (0.021)
IIGDM	0.389 (0.040)	0.89 (0.03)	8.78 (0.89)	0.214 (0.019)

Table 2: Diagnostics for predictive samples generated via elliptical slice sampling (ESS), DPS guidance, and IIGDM guidance, for the Poisson-count experiment of Section 4.1. Entries are means over 100 realisations, with standard deviation shown in parentheses.

Sampler	RMSE	90% coverage	ES	CRPS
ESS	0.547 (0.078)	0.89 (0.04)	12.35 (1.80)	0.304 (0.042)
DPS	0.574 (0.082)	0.84 (0.05)	13.00 (1.95)	0.320 (0.048)
IIGDM	0.552 (0.085)	0.89 (0.04)	12.48 (1.95)	0.306 (0.045)

ways: via ESS (200 independent chains of 500 iterations each, retaining the terminal state of each chain as one draw), and via guided reverse-time diffusion with 500 discretization steps using both DPS and IIGDM guidance. In Figure 2 we show results from the first realisation obtained using IIGDM, from which we see that the predictive samples recover both the marginal and the joint structure similarly to ESS. Table 2 summarises the diagnostics averaged over the 100 realisations. The DPS sampler retains the same mild under-dispersion seen in the linear-Gaussian case, while IIGDM closely matches ESS under these diagnostics.

Area-exceedance functional Here, we consider the area-exceedance data model (20) with threshold $c = 1$, variance $\xi^2 = 0.1$, sharpness $\kappa = 10$, and a single scalar observation $Z = 410$ (40% of the $n = 1024$ cells). For each sampler we draw 200 predictive samples: via guided reverse-time diffusion with 500 discretization steps under DPS and IIGDM guidance, and via ESS targeting the same discrete-Gaussian likelihood (200 independent chains of 500 iterations each, retaining the terminal state of each chain as one draw). Table 3 reports the realised exceedance counts, which are respected by all samplers, although we note that DPS under-disperses relative to ESS on the ‘hard’ count. Figure 3 shows four IIGDM realisations from the predictive distribution under the area-exceedance constraint.

When one has a latent Gaussian process model, such as in this section, there are various inferential methods that can be often used to effectively sample from the predictive distribution, such as ESS. The benefit of guided diffusion becomes clearer in the next section, where we apply it to non-Gaussian spatial process models.

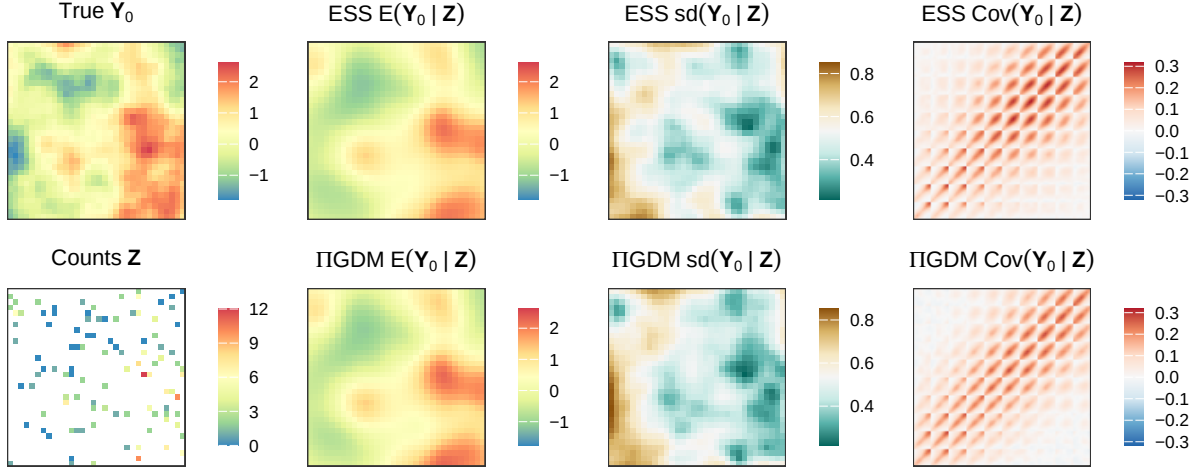


Figure 2: Predictive distributions under 10% Poisson-count observations from a Matérn 3/2 latent Gaussian process. Left: true field $\mathbf{Y}_0$ (top) and observed counts $\mathbf{Z}$ (bottom). Centre-left: predictive mean from ESS (top) and IIGDM (bottom). Centre-right: predictive standard deviation from ESS (top) and IIGDM (bottom). Right: predictive covariance of the central 10×10 block of grid cells from ESS (top) and IIGDM (bottom).

Table 3: Realised exceedance counts under the area-exceedance constraint $Z = 410$ ($c = 1$, $\kappa = 10$) for the experiment in Section 4.1. Each sampler draws 200 independent predictive realisations; entries report mean and standard deviation within parentheses for the exact count $\#(Y_{0,i} > c)$ and for the soft count $h(\mathbf{Y}_0) \equiv \sum_{i=1}^n \sigma_\kappa(Y_{0,i})$ targeted by the discrete-Gaussian surrogate (20) with $\xi^2 = 0.1$.

Sampler	$\#(Y_{0,i} > c)$	$h(\mathbf{Y}_0)$
ESS	410.2 (4.4)	410.0 (0.4)
DPS	409.6 (1.7)	410.0 (0.3)
IIGDM	408.5 (4.7)	410.0 (0.3)

4.2 Ginzburg–Landau process model

In this section we consider a non-Gaussian process model, for which no analytic distribution is available and for which the score function is hence intractable. Specifically, we consider the amplitude field $|A(\mathbf{s}, \tilde{t})|$, $\mathbf{s} \in [0, 40] \times [0, 40]$, $\tilde{t} \in [0, 20]$, of the two-dimensional complex Ginzburg–Landau equation (CGLE; Aranson and Kramer, 2002),

$$\begin{aligned} \frac{\partial A(\mathbf{s}, \tilde{t})}{\partial \tilde{t}} &= A(\mathbf{s}, \tilde{t}) + (1 + \iota c_1) \nabla^2 A(\mathbf{s}, \tilde{t}) - (1 - \iota c_3) |A(\mathbf{s}, \tilde{t})|^2 A(\mathbf{s}, \tilde{t}), \\ A(s_1 + L, s_2, \tilde{t}) &= A(s_1, s_2, \tilde{t}), \quad A(s_1, s_2 + L, \tilde{t}) = A(s_1, s_2, \tilde{t}), \end{aligned} \quad (21)$$

where ∇^2 is the Laplacian on a torus, $L = 40$ and $\iota \equiv \sqrt{-1}$. The CGLE is a canonical model for pattern formation in oscillatory media, arising in applications as diverse as fluid

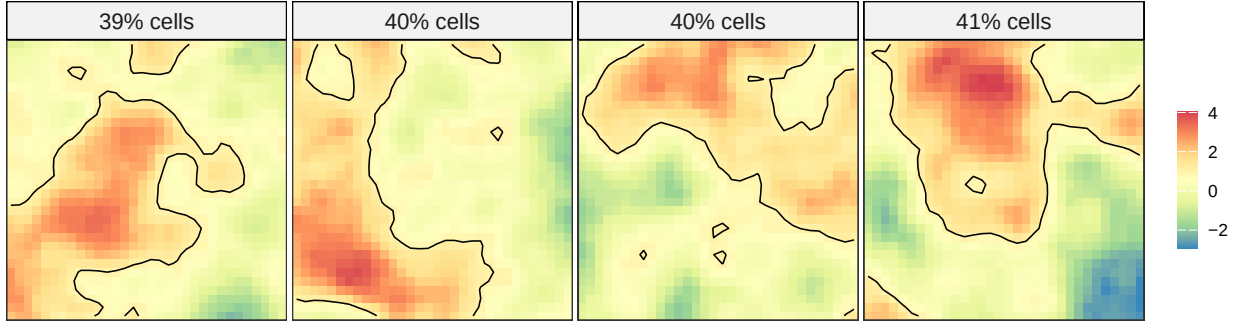


Figure 3: Four Π GDM predictive realisations under the single area-exceedance observation $Z = 410$ cells exceeding $c = 1$. The realisations shown are those at the 2nd, 34th, 66th and 98th percentiles of the realised exceedance count $\#\{Y_{0,i} > c\}$ across the 200 predictive draws. Each panel is annotated with the percentage of cells in that realisation that exceed the threshold; the black contour outlines the exceedance region $\{Y_0 > 1\}$.

dynamics and nonlinear optics, with solutions that include spiral waves and travelling fronts. We let $c_1 = c_3 = 1$ on a periodic 64×64 domain with sides of length 40. We solve (21), initialised at $\tilde{t} = 0$ to a small random complex-valued initial field, using the explicit Euler method with $\Delta\tilde{t} = 0.05$ up to physical time $\tilde{t} = 20$. We then compute $|A(\mathbf{s}, \tilde{t})|$ at $\tilde{t} = 20$, and construct $\mathbf{Y}_0$ by downsampling by a factor of two (so that $\mathbf{Y}_0$ has $n = 1024$ cells). We simulate from (21) in this way, with independent initial conditions, 8,000 times, to yield an ensemble that we split into 7,200 training and 800 held-out fields for training a neural network. Since this process model has no analytic density, the experiments below cannot be checked against an exact (or MCMC-based) predictive distribution; we therefore rely on predictive diagnostics throughout. Note that $\text{Cov}(\mathbf{Y}_0)$ is not available in closed form; for this reason we use $\mathbf{V}_t = \sigma_t^2 \mathbf{I}_n$ in (14) when approximating the noised likelihood (15) for Π GDM.

For $\mathbf{r}_\phi(\mathbf{Y}_t, t)$ we use a time-conditioned U-Net (Ronneberger et al., 2015) with circular padding, trained using the objective (7). We consider two data models: an areal linear-Gaussian aggregation model, and an upper-tail censoring model. For 100 different realisations of the ground truth, we generate synthetic observations under each data model, and we draw 200 predictive samples by discretising reverse diffusion with 500 steps. For both data models, we report RMSE, 90% coverage, ES, and CRPS, computed against the true fields and averaged over the 100 realisations.

Aggregated Gaussian observations In this example we define $\mathbf{H}$ in (16) so that $\mathbf{Z}$ is an 8×8 array of areal aggregates, corrupted with measurement error with standard deviation $\sigma_\epsilon = 0.02$. Figure 4 shows predictions from Π GDM guidance for one realisation. Table 4 reports the diagnostics averaged over 100 independent realisations. There is little difference between DPS and Π GDM in this example although, as we saw in the Gaussian-process experiments, Π GDM has better coverage properties. We further assess prediction quality through low-dimensional functionals of each predictive draw relative to the held-out truth. Specifically, we look at the global mean, the global minimum, the global maximum, the proportion of process values below 0.5, and the process in four grid cells (close to $(0, 0)$),

Table 4: Diagnostics from predictive sampling of the Ginzburg–Landau amplitude field under an areal-aggregation and an upper-tail censoring model. RMSE (of the predictive mean), 90%-interval coverage, ES, and CRPS are computed against the true field over all 1024 cells (in the aggregation example) or the censored cells (in the censoring example). Entries are means over 100 realisations, with standard deviation in parentheses.

Experiment	Sampler	RMSE	90% coverage	ES	CRPS
Areal aggregates	DPS	0.091 (0.023)	0.855 (0.037)	2.05 (0.53)	0.045 (0.012)
	IIGDM	0.093 (0.021)	0.886 (0.029)	2.09 (0.49)	0.046 (0.011)
Censoring ($c = 1$)	DPS	0.010 (0.002)	0.828 (0.049)	0.10 (0.01)	0.005 (0.001)
	IIGDM	0.011 (0.002)	0.861 (0.051)	0.10 (0.02)	0.006 (0.001)

(13, 27), (27, 13) and (39, 39)). Figure S.2 of the Supplementary Material summarises the differences between the true underlying functionals and the sampled functionals for 100 different realisations, with DPS and IIGDM under areal aggregation. The differences are centred near zero with a small spread across all functionals for both samplers, indicating good recovery of these functionals with a slight bias in the predicted global mean.

Censored data In this example we assume that the measurements are censored in their upper tail at $c = 1$ and that the uncensored cells are observed with measurement error having standard deviation $\sigma_\epsilon = 0.03$. Note that, because of measurement error, a cell may be censored even when its underlying process value lies below c , and vice versa. Figure 5 shows predictions from IIGDM guidance for one realisation. Table 4 reports the diagnostics averaged over 100 independent realisations with, again, IIGDM achieving closer-to-nominal coverage than DPS. Figure S.3 of the Supplementary Material shows the differences of the functionals analysed in the areal-observation case. The predictions again appear to be consistent with the underlying truth, with the global mean for IIGDM showing a small positive bias while the remaining functional differences stay close to zero.

Note that both experiments in this section used the same neural network for estimating the process model score function; only the guidance gradient differed between the two. With NCS, a separate network would need to be trained for each data model. Section S.2 of the Supplementary Material illustrates potential problems that could arise if NCS is used in a setting where the data model is misspecified.

5 Observing system simulation experiment of methane-flux inversion over the UK and Ireland

We now explore the potential of guided diffusion in a realistic controlled setting, by applying it to atmospheric methane-flux inversion over the UK and Ireland. The particular inversion problem we consider can be tackled using MCMC which, however, scales poorly with spatial dimension. We show that, here, guided diffusion can be used to obtain predictive inferences that are similar to those from MCMC, while offering the possibility to scale up to higher

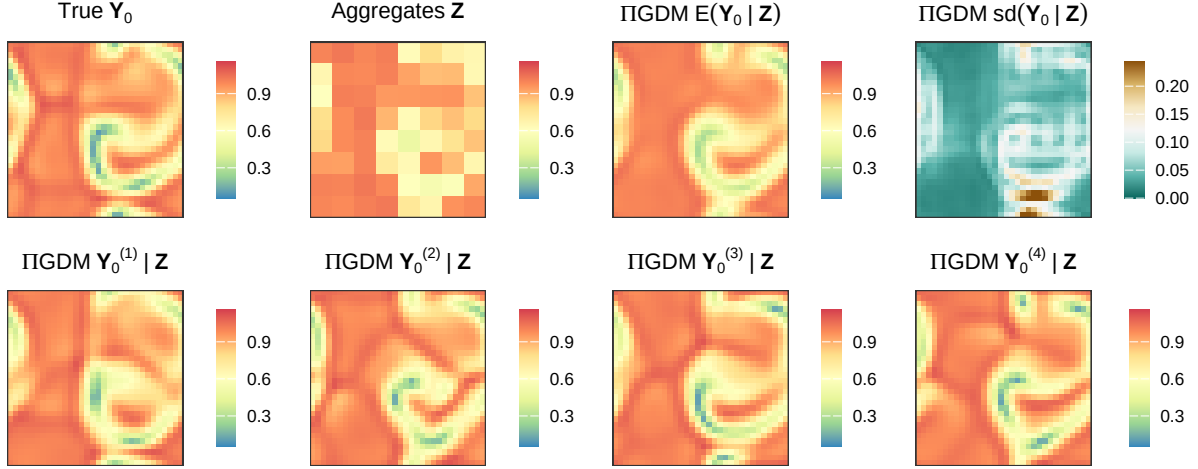


Figure 4: Guided IIGDM predictions of a Ginzburg–Landau amplitude field under 64 areal aggregates over 4×4 blocks. Top: true field, the block averages, the predictive mean and the predictive standard deviation obtained using IIGDM. Bottom: four draws from the predictive distribution using IIGDM.

spatial resolutions.

5.1 Trace-gas inversion using guided diffusion

The specific atmospheric trace-gas inversion problem we consider is to infer a spatial flux field $\mathbf{Y}_0 \equiv (Y_0(\mathbf{s}) : \mathbf{s} \in \mathcal{G})^\top$ on a gridded domain $\mathcal{G}$ from observations of the atmospheric mole fraction recorded at a small number of ground stations. For a long-lived gas such as methane, the mole fraction is, to first order, a known linear functional of the flux: meteorology and transport map the flux at every grid cell to the mole fraction at each station and time point through a sensitivity (“footprint”) operator computed by a Lagrangian particle-dispersion model (here we use the UK Met Office Numerical Atmospheric-dispersion Modelling Environment, NAME; Jones et al., 2007). The data of methane mole fraction $\mathbf{Z} \in \mathbb{R}^m$, which are recorded at stations around the UK at fine time intervals, are assumed to follow the data model

$$\mathbf{Z} \mid \mathbf{Y}_0 \sim \text{Gau}(\mathbf{H}\mathbf{Y}_0, \boldsymbol{\Sigma}_\epsilon), \quad \boldsymbol{\Sigma}_\epsilon = \boldsymbol{\Sigma}_\zeta + \boldsymbol{\Sigma}_o, \quad (22)$$

where $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the (NAME-derived) transport operator mapping the n flux cells to the m station–time observations, $\boldsymbol{\Sigma}_\zeta \in \mathbb{R}^{m \times m}$ is a covariance matrix accounting for transport-model error, and $\boldsymbol{\Sigma}_o$ is a diagonal measurement-error covariance matrix. For illustration purposes, here we assume that the transport-model error covariance is known from previous studies. Following Zammit-Mangion et al. (2016), we take $\boldsymbol{\Sigma}_\zeta$ to be a space-time separable covariance matrix, specified through $\boldsymbol{\Sigma}_\zeta^{-1} = \sigma_\zeta^{-2} (\mathbf{Q}_a^t \otimes (\mathbf{R}_d^s)^{-1})$, where $\mathbf{Q}_a^t$ is the precision of a first-order autoregressive process in time with lag-one correlation $a = 0.9$ on the two-hourly observation grid, $\mathbf{R}_d^s$ is an exponential spatial correlation matrix across the stations with range $d = 2.5^\circ$, and $\sigma_\zeta = 10$ ppb is a scale parameter. The measurement-error covariance

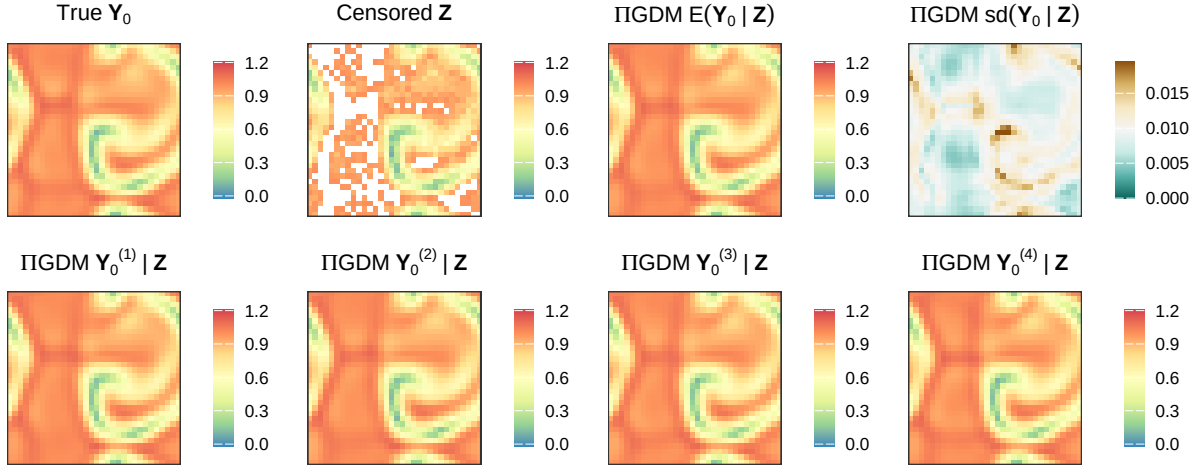


Figure 5: Guided IIGDM predictions of a Ginzburg–Landau amplitude field under upper-tail censoring at $c = 1$. Top: true field, the uncensored observations (with censored cells shown in white), the predictive mean and the predictive standard deviation. Bottom: four draws from the predictive distribution using IIGDM.

Σ_o is diagonal with diagonal elements 1 ppb². Model (22) is the linear-Gaussian model of Section 3.2, but with a non-diagonal covariance matrix $\text{Cov}(\epsilon) = \Sigma_\epsilon$.

The first setup we consider is the same as that in Zammit-Mangion et al. (2016). The gridded flux domain $\mathcal{G}$ is a $0.7^\circ \times 0.5^\circ$ longitude–latitude grid over the UK and Ireland with $n = 122$ land cells (arising from downsampling the native transport-model resolution by a factor of two). The true (but assumed unknown) flux field is taken to be the UK and Ireland methane inventory (a combination of the NAEI and EDGAR inventories; Ganesan et al., 2015), whose domain total at this resolution is 2.73 Tgyr^{-1} ($86,590 \text{ gs}^{-1}$). We use the transport model NAME to construct the matrix $\mathbf{H}$, so that the (i, j) th element of $\mathbf{H}$ corresponds to the sensitivity of the i th observation to the flux in the j th grid cell (sensitivities are computed by running NAME backwards in time for each space-time observation; see Ganesan et al., 2015, for details). We then forward-simulate mole-fraction observations through (22) at four stations: Mace Head, Ireland (MHD); Ridge Hill, England (RGL); Tacolneston, England (TAC); and Angus, Scotland (TTA). Simulations are done at two-hourly resolution over January–March 2014 using the true meteorological conditions during that period. A missingness pattern corresponding to that in actual data is applied, giving a total of $m = 3409$ space-time observations. The domain, station footprints, and simulated time series are shown in Figure 6 (figure adapted from Zammit-Mangion et al. 2016).

Following Zammit-Mangion et al. (2016), we model the flux field as a Box–Cox spatial process (De Oliveira et al., 1997). Let

$$g_\lambda(y) \equiv \begin{cases} (y^\lambda - 1)/\lambda, & \lambda \neq 0, \\ \log y, & \lambda = 0, \end{cases} \quad (23)$$

denote the Box–Cox transformation. We model $g_\lambda(\mathbf{Y}_0)$ as a truncated Gaussian vector with

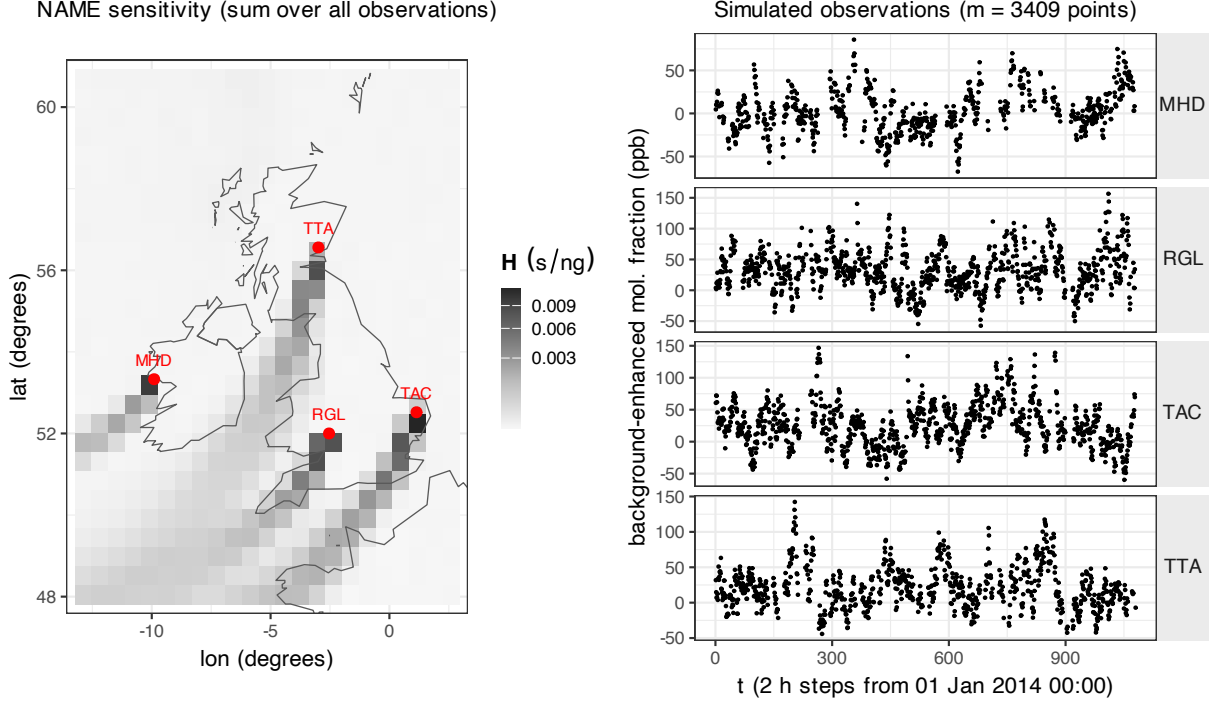


Figure 6: Set-up for the methane controlled experiment. Left: the UK and Ireland flux domain with the four station locations (MHD, RGL, TAC, TTA) and the NAME sensitivity plumes comprising $\mathbf{H}$ at the first time point (corresponding to January 01 2014 at 00:00). Right: the four simulated ‘above background’ methane mole-fraction time series ($m = 3409$ observations, with a sampling interval of 2 h) used as $\mathbf{Z}$ in Section 5.

density

$$p(g_\lambda(\mathbf{Y}_0) \mid \mathbf{b}, \tau, \boldsymbol{\theta}, \lambda) \propto \exp\left(-\frac{\tau}{2} (g_\lambda(\mathbf{Y}_0) - \mathbf{X}\mathbf{b})^\top \mathbf{R}_\theta^{-1} (g_\lambda(\mathbf{Y}_0) - \mathbf{X}\mathbf{b})\right) T_\lambda(g_\lambda(\mathbf{Y}_0)), \quad (24)$$

and express the density of $\mathbf{Y}_0$ as $\mathbf{Y}_0 \sim \text{BC}(\mathbf{X}\mathbf{b}, \tau^{-1}\mathbf{R}_\theta, \lambda)$. Here, $T_\lambda(\tilde{y}) \equiv 1(\tilde{y} > -1/\lambda)$ for $\lambda > 0$, and $1(\tilde{y} < -1/\lambda)$ for $\lambda < 0$ and, like $g_\lambda(\cdot)$, is applied elementwise to its vector argument. The matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ contains covariates, $\mathbf{b} \in \mathbb{R}^p$ is the corresponding vector of regression coefficients, τ is a marginal precision parameter, and $\mathbf{R}_\theta \equiv (\exp(-\theta_1 \|\mathbf{s}_i - \mathbf{s}_j\|^{\theta_2}) : i, j = 1, \dots, n)$ is a powered-exponential correlation matrix with $\theta_1 > 0$ and $0 < \theta_2 < 2$. As covariates we use two indicator functions that split the domain at 56.4°N , to allow for the much lower emissions typical of the Scottish Highlands; the two elements of $\mathbf{b}$ are therefore the mean levels of $g_\lambda(\mathbf{Y}_0)$ outside and inside the Highlands, respectively.

We equip the unknown parameters in the flux field with prior distributions. Since $\mathbf{b}$ and τ are defined on the Box–Cox scale, which changes with λ , these parameters cannot be assigned independent prior distributions. We therefore instead assign prior distributions to these parameters on their natural scale. Specifically, we define a prior distribution for the

transformed parameters:

$$\tilde{b}_1 = g_\lambda^{-1}(b_1), \quad \tilde{b}_2 = g_\lambda^{-1}(b_2), \quad c_V = \tau^{-1/2}(\tilde{b}_1^\lambda)^{-1} \quad (25)$$

where the expression for the coefficient of variation c_V follows from

$$\text{sd}(g_\lambda(Y_0(\mathbf{s}))) \approx \tilde{b}_1^{\lambda-1} \text{sd}(Y_0(\mathbf{s})) = \tilde{b}_1^\lambda c_V.$$

We elicit the prior distribution by first running MCMC on the EDGAR/NAEI inventory treated as a realisation of $\mathbf{Y}_0$, under uninformative priors for the parameters (MCMC is computationally tractable when $n = 122$). We use the posterior distribution of the parameters from that run to construct a prior distribution for the flux inversion from mole fraction data. Figure S.4 of the Supplementary Material shows these elicited parameter prior densities. The full hierarchical model we use in the inversion is given by

Data model	$(\mathbf{Z} \mid \mathbf{Y}_0) \sim \text{Gau}(\mathbf{H}\mathbf{Y}_0, \Sigma_\epsilon),$
Process model	$(\mathbf{Y}_0 \mid \mathbf{b}, \tau, \boldsymbol{\theta}, \lambda) \sim \text{BC}(\mathbf{X}\mathbf{b}, \tau^{-1}\mathbf{R}_\theta, \lambda),$
Parameter model	$\lambda \sim \text{Gau}(0.168, 0.068^2) I(0 < \lambda < 0.5),$ $\log \theta_1 \sim \text{Gau}(-0.005, 0.387^2), \quad \text{logit}(\theta_2/2) \sim \text{Gau}(0.290, 0.456^2),$ $\log \tilde{b}_1 \sim \text{Gau}(6.334, 0.260^2), \quad \log \tilde{b}_2 \sim \text{Gau}(4.540, 0.464^2),$ $\log c_V \sim \text{Gau}(-0.366, 0.168^2),$

(26)

with $\mathbf{b}$ and τ obtained from $(\lambda, \tilde{b}_1, \tilde{b}_2, c_V)$ through (25). The logarithmic and logit transformations respect the parameter constraints: $\tau, \theta_1 > 0$ and $\theta_2 \in (0, 2)$.

Marginalising (26) over $(\mathbf{b}, \tau, \boldsymbol{\theta}, \lambda)$ yields a non-Gaussian process model $p(\mathbf{Y}_0)$ that is available only implicitly through simulation: We draw parameters from the parameter prior distribution, and then draw $\mathbf{Y}_0$ from the corresponding Box–Cox process in order to provide a training data set of size 3.6×10^4 for training the score network. The mole-fraction data model (22) is not used for training, it only enters through the guidance gradient when making inference. Figure S.5 of the Supplementary Material shows unconditional draws from the trained generative model.

5.2 Guided predictions and comparison with MCMC

We draw a sample of size 500 from the predictive distribution using guided diffusion with 500 discretization steps (setting $\mathbf{V}_t = \sigma_t^2 \mathbf{I}_n$ after data standardisation for IIGDM), and compare it to the Hamiltonian-Monte-Carlo-within-Gibbs sampling of $\mathbf{Y}_0$ and the model parameters in (26) using the code of Zammit-Mangion et al. (2016). Table 5 shows that both guided predictions approximately match MCMC on several diagnostics computed against the truth: RMSE, per-grid-cell correlation, CRPS, and ES. The 90%-interval coverages are similar across the three samplers, and all three domain-total estimates are biased slightly upward, with the MCMC total closest to the truth. The predictive mean and standard deviation maps in Figure 7 are visually almost indistinguishable across the three samplers, and the guided and MCMC predictions agree more closely with each other than either agrees with the truth; see Figure 8.

Table 5: Methane flux-field diagnostics for the two guided samplers (DPS and Π GDM) and MCMC targeting the same hierarchical model (26) (Section 5.2). RMSE, correlation, 90%-interval coverage, CRPS, and ES are computed against the true flux field over the $n = 122$ cells; the final column gives the domain-total predictive mean and predictive standard deviation. The true domain total is $86,590 \text{ g s}^{-1}$ (2.73 Tg yr^{-1}); coverage target is 0.90. The diagnostics are computed from a sample of size 500.

Method	RMSE (g s^{-1})	corr (truth)	90% cov.	CRPS	ES	total $\pm$ sd (g s^{-1})
DPS (reverse SDE)	298	0.80	0.80	161	2,336	$87,318 \pm 2,547$
Π GDM (reverse SDE)	296	0.80	0.82	159	2,318	$87,367 \pm 2,938$
MCMC	297	0.80	0.80	158	2,336	$86,644 \pm 3,015$

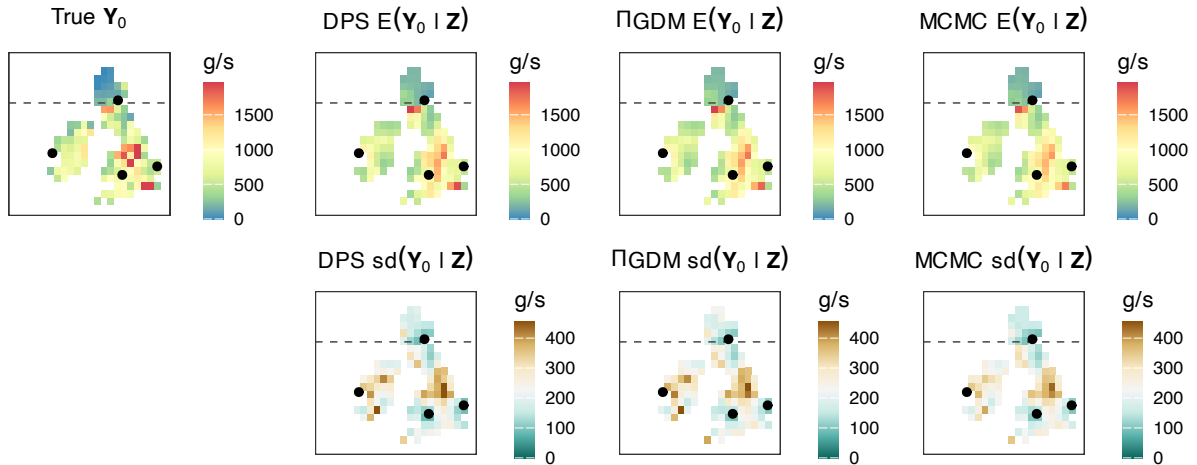


Figure 7: Predictive mean and predictive standard deviation of the methane flux field for DPS, Π GDM, and MCMC (g s^{-1} , $n = 122$). The leftmost panel shows the methane flux inventory map, here treated as the underlying truth. Station locations are marked with dots, while the dashed line is the 56.4°N covariate split.

Figure 9 compares the marginal predictions of the guided and MCMC samplers at twelve randomly selected cells. The predictive distributions are closely matched in location, spread, and skewness at every cell. Table S.2 of the Supplementary Material reports region-level totals. The guided (DPS and Π GDM) and MCMC predictive distributions agree closely with one another: all three reproduce the English and Welsh totals, over-estimate the Scottish total, and slightly under-estimate the Northern Irish and Irish totals. These territorial biases sum to the domain-total offset, indicating a coherent spatial redistribution of flux, as is typically seen in studies of this type.

We assess prediction quality through low-dimensional functionals of the predictive distribution. Figure 10 compares, for DPS, Π GDM, and MCMC, the domain total and maximum,

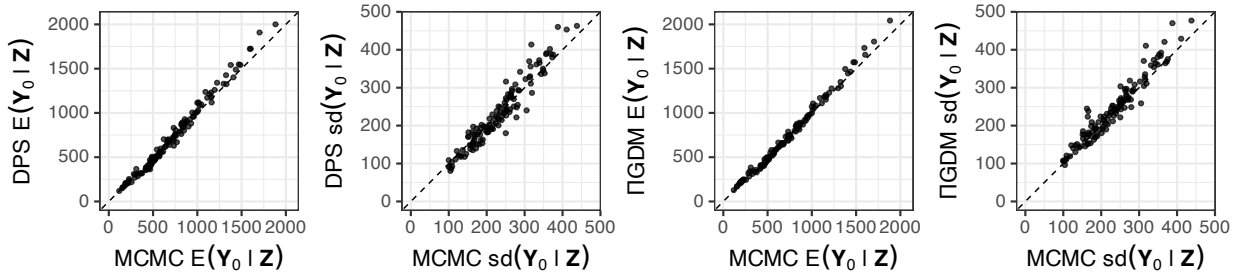


Figure 8: Per-cell agreement between each guided sampler and MCMC over the 122 cells. From left to right: DPS predictive means, DPS predictive standard deviations, Π GDM predictive means, and Π GDM predictive standard deviations. Dashed lines indicate the 1:1 reference.

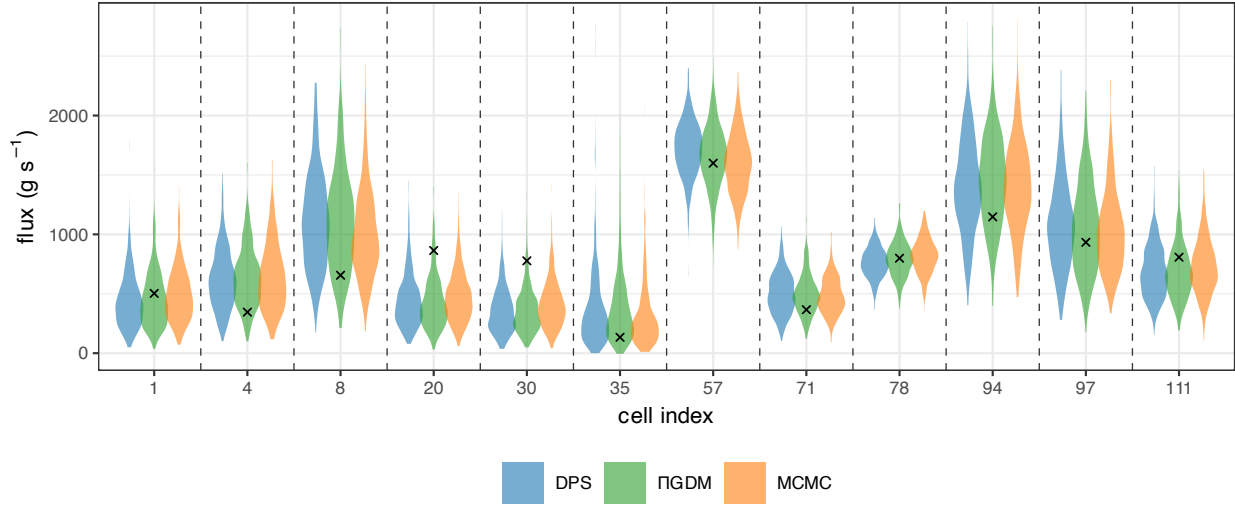


Figure 9: Marginal predictive distributions of the flux (g s^{-1}) at twelve randomly chosen grid cells, comparing the guided samplers (blue and green) with MCMC (orange). The black cross marks the true (inventory) flux.

the spatial 5th and 95th percentiles, and the marginal flux at four cities. The guided and MCMC distributions corroborate on all quantities except the upper percentile where the approximation done with guided diffusion is likely leading to slight over-estimation of the fluxes.

5.3 Computational efficiency

Implementing MCMC is possible in this application where the data model is Gaussian, and where the hierarchical model can be sampled from using Gibbs steps. The utility of guidance

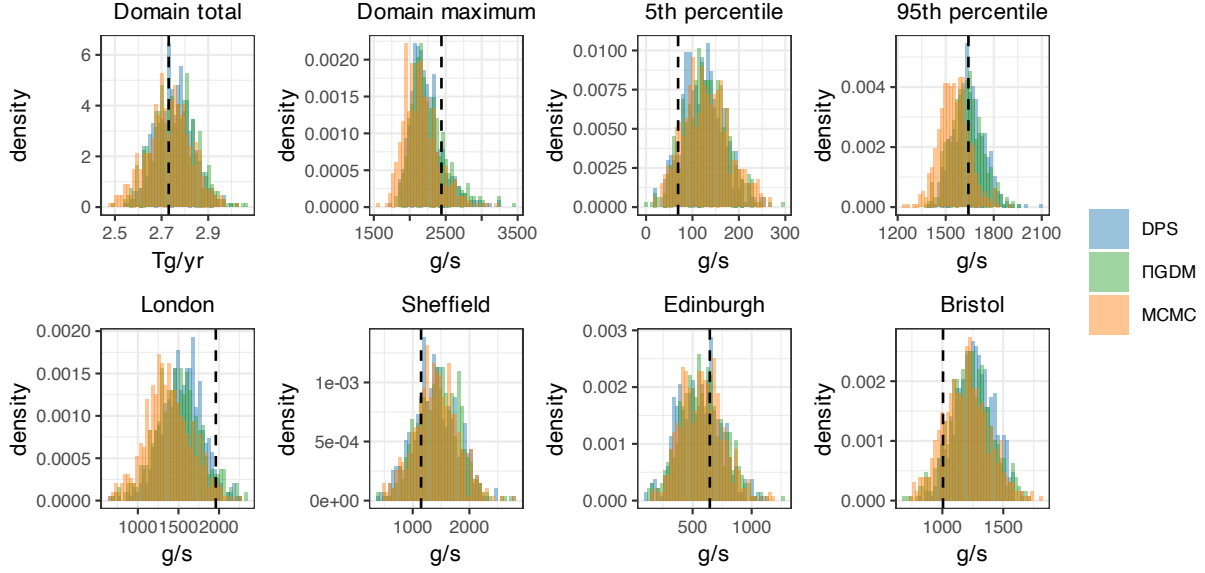


Figure 10: Predictive distributions of eight low-dimensional functionals of the methane flux field ($n = 122$), for DPS (blue), IIGDM (green), and MCMC (orange, labelled MCMC); the dashed line marks the true (inventory) value. Top row: domain total, domain maximum, and the within-draw spatial 5th and 95th percentiles. Bottom row: marginal flux at grid cells nearest to London, Sheffield, Edinburgh, and Bristol.

here lies in speed and scalability: the score needed in the reverse diffusion process is estimated using feedforward operations and has cost that scales linearly with the dimension of $\mathbf{Y}_0$. On the other hand, MCMC and other Bayesian algorithms applied to spatial processes of this kind require factorising a matrix of size $n \times n$ several times. To assess scalability, we consider the seconds per effective sample of the total methane flux for both approaches under different spatial resolutions: $n = 122$ (the downsampled case considered above), $n = 487$ (the native resolution), and $n = 1948$, where the flux map and the sensitivity plumes at this higher resolution are obtained through piecewise linear interpolation. Each resolution has its own simulated mole-fraction data $\mathbf{Z}$, generated from the flux field at that resolution under the same transport-error model (22); the number of observations $m = 3409$ is held fixed across the three cases. At the higher resolutions we use the same model in (26) but with $\tilde{b}_1$ and $\tilde{b}_2$ appropriately rescaled since they describe median fluxes in a single grid cell, and hence need to be adapted to the cell area. For guided diffusion, every sample is an effective sample (albeit from an approximate posterior distribution) while, for MCMC, chain correlation is taken into account when calculating effective sample size. We run guided diffusion on an NVIDIA H100 GPU, and MCMC with ten chains in parallel on ten Xeon(R) Gold 6330 CPU cores.

IIGDM required 0.071, 0.234 and 0.506 seconds per effective sample of the domain total flux for $n = 122$, 487 and 1948, respectively. MCMC required 0.010, 0.353 and 5.45 seconds, respectively. Therefore, MCMC is more computationally efficient in low dimension, but

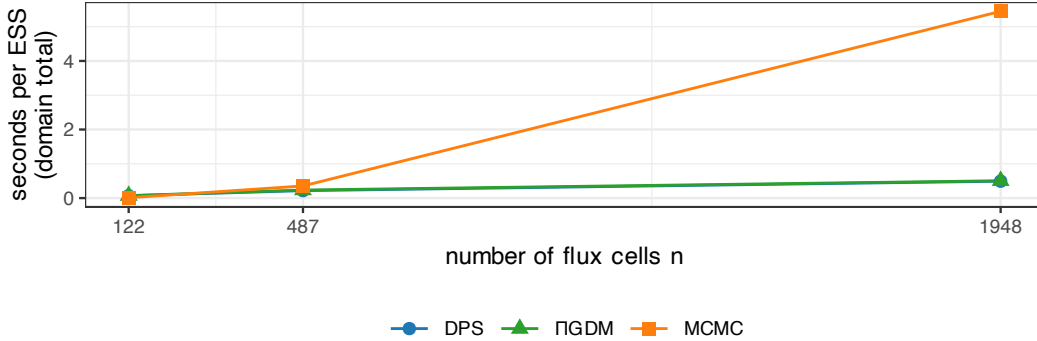


Figure 11: Computational scaling of guided diffusion and MCMC for the methane inversion shown as the time required per effective sample of domain total against the number of flux cells n , where $n = 122, 487$ and 1948 . Guided-diffusion timings are from running reverse diffusion on a GPU, while MCMC timings are from running ten parallel chains on ten CPU cores.

its cost grows dramatically faster than that of guided prediction because of the matrix factorisation required at each iteration; it is roughly eleven times more expensive per effective sample at $n = 1948$, and it is not practical at higher resolutions. Figure 11 shows the time required per effective sample of the domain total flux against n for both methods, and clearly shows the potential of guided diffusion in high-resolution flux inversion. Functional diagnostics and predictions at these higher resolutions are shown in Figures S.6–S.7 and S.8–S.9 of the Supplementary Material.

6 Discussion

In this paper we apply guided diffusion to spatial prediction, and show its applicability to a wide variety of process and data models that include right censoring and area-exceedance functionals. Where feasible, we show that guided diffusion recovers predictive distributions that are close to what one can obtain from MCMC. For the field of spatial statistics, the implications are threefold. First, the method is very general, and can be applied whenever one is able to simulate from the process model and when one can construct a differentiable likelihood function from a data model. Second, the method lends itself to *amortisation*: once a neural network is trained to emulate a process, it can be re-used repeatedly under very different data-model-scenarios. Third, even when generality and amortisation are not a concern, the method is still useful for scalability reasons. As shown in Section 5.3, guided diffusion will in many cases scale linearly with the dimension of the process. MCMC, on the other hand, often scales poorly with dimension.

There are some limitations of the proposed approach to spatial prediction. First, we found guided diffusion to approximate MCMC fairly well, with IIGDM achieving coverage closer to nominal than DPS on the cases we considered. However, both are slightly under-dispersed when compared to MCMC and both involve several approximations that, unlike MCMC, do not have any notion of asymptotic exactness. Second, vanilla guided diffusion

freezes the process model, the process-model prior distributions, and the gridding of $\mathcal{D}$, and unlike MCMC, returns no posterior distribution over the process-model parameters θ . This may be unattractive in applications where the process-model parameters carry physical meaning. An appealing way forward is to train the “prior” diffusion model over both the process and the parameters (Gloeckler et al., 2024) or to combine guided diffusion with neural parameter estimation in an expectation-maximisation style framework (Zammit-Mangion et al., 2025; Sainsbury-Dale et al., 2025). Finally, in the non-Gaussian case, the framework relies on having sufficient realisations from the process model in order to construct a network to estimate the score function required for reverse diffusion. For some applications where simulation is expensive (e.g., full Navier–Stokes type equations), this may be prohibitive. An alternative in this setting is to avoid score estimation altogether: Sharrock et al. (2026) write the process as a deterministic map of a latent innovation with a tractable reference distribution, so that conditioning requires solving a single reverse-time SDE and no training at all. This is attractive when simulations are scarce, although it requires the map from innovation to process to be available and differentiable.

Acknowledgements

The authors would like to thank Alistair Manning for supplying the NAME simulations used in Section 5.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used Claude in order to help produce code and documentation from preliminary scripts and prompts. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- Anderson, B. D. O. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.
- Aranson, I. S. and Kramer, L. (2002). The world of the complex Ginzburg–Landau equation. *Reviews of Modern Physics*, 74(1):99–143.
- Bertolacci, M., Zammit-Mangion, A., Schuh, A., Bukosa, B., Fisher, J. A., Cao, Y., Kaushik, A., and Cressie, N. (2024). Inferring changes to the global carbon cycle with WOMBAT v2.0, a hierarchical flux-inversion framework. *The Annals of Applied Statistics*, 18(1):303–327.
- Bolin, D. and Lindgren, F. (2015). Excursion and contour uncertainty regions for latent Gaussian models. *Journal of the Royal Statistical Society B*, 77(1):85–106.

- Byrne, B., Baker, D. F., Basu, S., Bertolacci, M., Bowman, K. W., Carroll, D., Chatterjee, A., Chevallier, F., Ciais, P., Cressie, N., et al. (2023). National CO₂ budgets (2015–2020) inferred from atmospheric CO₂ observations in support of the global stocktake. *Earth System Science Data*, 15(2):963–1004.
- Cardoso, G., El Idrissi, Y. J., Le Corff, S., and Moulines, E. (2024). Monte Carlo guided denoising diffusion models for Bayesian linear inverse problems. In *International Conference on Learning Representations*. Online: <https://openreview.net/forum?id=nHESwXvxWK>.
- Chung, H., Kim, J., McCann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*. Online: <https://openreview.net/forum?id=0nD9zGAGT0k>.
- Cseke, B., Zammit-Mangion, A., Heskes, T., and Sanguinetti, G. (2016). Sparse approximate inference for spatio-temporal point process models. *Journal of the American Statistical Association*, 111(516):1746–1763.
- De Oliveira, V., Kedem, B., and Short, D. A. (1997). Bayesian prediction of transformed Gaussian random fields. *Journal of the American Statistical Association*, 92(440):1422–1433.
- Dombry, C., Eyi-Minko, F., and Ribatet, M. (2013). Conditional simulation of max-stable processes. *Biometrika*, 100(1):111–124.
- Efron, B. (2011). Tweedie’s formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614.
- Ganesan, A. L., Manning, A. J., Grant, A., Young, D., Oram, D. E., Sturges, W. T., Moncrieff, J. B., and O’Doherty, S. (2015). Quantifying methane and nitrous oxide emissions from the UK and Ireland using a national-scale monitoring network. *Atmospheric Chemistry and Physics*, 15(11):6393–6406.
- Gloeckler, M., Deistler, M., Weilbach, C. D., Wood, F., and Macke, J. H. (2024). All-in-one simulation-based inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 15735–15766. PMLR.
- Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.
- Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates Inc.
- Jones, A., Thomson, D., Hort, M., and Devenish, B. (2007). The U.K. Met Office’s next-generation atmospheric dispersion model, NAME III. In Borrego, C. and Norman, A.-L., editors, *Air Pollution Modeling and Its Application XVII*, pages 580–589. Springer, Boston, MA.

- Kawar, B., Elad, M., Ermon, S., and Song, J. (2022). Denoising diffusion restoration models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23593–23606. Curran Associates, Inc.
- Lai, C.-H., Song, Y., Kim, D., Mitsufuji, Y., and Ermon, S. (2025). The principles of diffusion models. *arXiv preprint arXiv:2510.21890*.
- Martín-Español, A., Zammit-Mangion, A., Clarke, P. J., Flament, T., Helm, V., King, M. A., Luthcke, S. B., Petrie, E., Rémy, F., Schön, N., et al. (2016). Spatial and temporal Antarctic Ice Sheet mass trends, glacio-isostatic adjustment, and surface processes from a joint inversion of satellite altimeter, gravity, and GPS data. *Journal of Geophysical Research: Earth Surface*, 121(2):182–200.
- Moss, H., Astfalck, L., Cowperthwaite, T., Doumont, C., Willis, S., Hennig, P., Nemeth, C., and Zammit-Mangion, A. (2026). Conditioning Gaussian processes on almost anything. *arXiv preprint arXiv:2605.21041*.
- Murray, I., Adams, R. P., and MacKay, D. J. C. (2010). Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548.
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.
- Rozet, F. and Louppe, G. (2023). Score-based data assimilation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 40521–40541. Curran Associates, Inc.
- Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society B*, 71(2):319–392 (with Discussion).
- Sainsbury-Dale, M., Zammit-Mangion, A., Cressie, N., and Huser, R. (2025). Neural parameter estimation with incomplete data. *arXiv preprint arXiv:2501.04330*.
- Sharrock, L., Astfalck, L., and Moss, H. (2026). LatentFlow: A general framework for conditioning stochastic processes. *arXiv preprint arXiv:2607.12922*.
- Sisson, S. A., Fan, Y., and Beaumont, M., editors (2018). *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, Boca Raton, FL.

- Song, J., Vahdat, A., Mardani, M., and Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*. Online: https://openreview.net/forum?id=9_gsMA8MRKQ.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*. Online: <https://openreview.net/forum?id=PXTIG12RRHS>.
- Sweet, W. V., Kopp, R. E., Weaver, C. P., Obeysekera, J., Horton, R. M., Thieler, E. R., and Zervas, C. (2017). Global and regional sea level rise scenarios for the United States. Technical Report NOAA Technical Report NOS CO-OPS 083, National Oceanic and Atmospheric Administration, Center for Operational Oceanographic Products and Services.
- Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.
- Walchessen, J., Zammit-Mangion, A., Huser, R., and Kuusela, M. (2025). Neural conditional simulation for complex spatial processes. *arXiv preprint arXiv:2508.20067*.
- White, P. A., Frye, H. A., Slingsby, J. A., Silander Jr, J. A., and Gelfand, A. E. (2024). Generative spatial generalized dissimilarity mixed modelling (spGDMM): An enhanced approach to modelling beta diversity. *Methods in Ecology and Evolution*, 15(1):214–226.
- Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 31372–31403.
- Zammit-Mangion, A., Bertolacci, M., Fisher, J., Stavert, A., Rigby, M., Cao, Y., and Cressie, N. (2022). WOMBAT v1.0: a fully Bayesian global flux-inversion framework. *Geoscientific Model Development*, 15(1):45–73.
- Zammit-Mangion, A., Cressie, N., and Ganesan, A. L. (2016). Non-Gaussian bivariate modelling with application to atmospheric trace-gas inversion. *Spatial Statistics*, 18(B):194–220.
- Zammit-Mangion, A., Rougier, J., Schön, N., Lindgren, F., and Bamber, J. (2015). Multivariate spatio-temporal modelling for assessing Antarctica’s present-day contribution to sea-level rise. *Environmetrics*, 26(3):159–177.
- Zammit-Mangion, A., Sainsbury-Dale, M., and Huser, R. (2025). Neural methods for amortized inference. *Annual Review of Statistics and Its Application*, 12(1):311–335.

A Derivations of (3), (4) and (6)

For completeness, here we give derivations of the conditional distribution (3), the reverse-time SDE (4), and the denoising score-matching objective (6). We obtain these from Anderson (1982); Vincent (2011); Song et al. (2021), but re-write them here in the notation and context of Section 2.

A.1 Conditional distribution (3) and the coefficients α_t and σ_t^2

The forward diffusion (2) is a linear-Gaussian SDE with time-dependent coefficients. Conditional on $\mathbf{Y}_0$ the solution to the SDE is therefore an affine functional of Brownian motion, and it therefore suffices to compute its mean and variance.

Consider the integrating factor $\alpha_t^{-1} = \exp(\frac{1}{2} \int_0^t \beta_s ds)$, so that $\frac{d}{dt} \alpha_t^{-1} = \frac{1}{2} \beta_t \alpha_t^{-1}$. Itô's product rule for a deterministic function α_t^{-1} and stochastic process $\mathbf{Y}_t$ is given by $d(\alpha_t^{-1} \mathbf{Y}_t) = (d\alpha_t^{-1}) \mathbf{Y}_t + \alpha_t^{-1} (d\mathbf{Y}_t)$. Therefore,

$$d(\alpha_t^{-1} \mathbf{Y}_t) = \frac{1}{2} \beta_t \alpha_t^{-1} \mathbf{Y}_t dt + \alpha_t^{-1} \left(-\frac{1}{2} \beta_t \mathbf{Y}_t dt + \sqrt{\beta_t} d\mathbf{W}_t \right) = \alpha_t^{-1} \sqrt{\beta_t} d\mathbf{W}_t. \quad (\text{A.1})$$

Integrating (A.1) from 0 to t and multiplying through by α_t gives the explicit solution

$$\mathbf{Y}_t = \alpha_t \mathbf{Y}_0 + \alpha_t \int_0^t \alpha_s^{-1} \sqrt{\beta_s} d\mathbf{W}_s. \quad (\text{A.2})$$

Since the Itô integral has zero mean, taking conditional expectations in (A.2) yields

$$\mathbb{E}[\mathbf{Y}_t \mid \mathbf{Y}_0] = \alpha_t \mathbf{Y}_0, \quad (\text{A.3})$$

which is the mean in (3). For the variance, note that the Itô integral in (A.2) has independent components. We can therefore apply the Itô isometry to each of these components to see that $\text{Cov}(\mathbf{Y}_t \mid \mathbf{Y}_0) = \sigma_t^2 \mathbf{I}_n$ with

$$\sigma_t^2 = \alpha_t^2 \int_0^t \alpha_s^{-2} \beta_s ds. \quad (\text{A.4})$$

Since $\alpha_s^{-2} = \exp(\int_0^s \beta_u du)$ satisfies $\frac{d}{ds} \alpha_s^{-2} = \beta_s \alpha_s^{-2}$, we have that

$$\int_0^t \alpha_s^{-2} \beta_s ds = \int_0^t \frac{d}{ds} \alpha_s^{-2} ds = \alpha_t^{-2} - \alpha_0^{-2} = \alpha_t^{-2} - 1, \quad (\text{A.5})$$

using $\alpha_0 = 1$. Substituting (A.5) into (A.4) gives

$$\sigma_t^2 = \alpha_t^2 (\alpha_t^{-2} - 1) = 1 - \alpha_t^2,$$

which is the variance in (3).

A.2 Reverse-time SDE (4)

Write the forward diffusion (2) in the generic Itô form,

$$d\mathbf{Y}_t = f(\mathbf{Y}_t, t) dt + g(t) d\mathbf{W}_t, \quad (\text{A.6})$$

where $f(\mathbf{Y}_t, t) \equiv -\frac{1}{2}\beta_t \mathbf{Y}_t$, and $g(t) \equiv \sqrt{\beta_t}$. Let $p(\cdot)$ denote the marginal density of $\mathbf{Y}_t$. By the Fokker–Planck (forward) equation, $p(\cdot)$ evolves according to

$$\partial_t p(\mathbf{y}_t) = -\nabla \cdot [f(\mathbf{y}_t, t) p(\mathbf{y}_t)] + \frac{1}{2} g(t)^2 \nabla^2 p(\mathbf{y}_t), \quad (\text{A.7})$$

where $\nabla \cdot$ and ∇^2 denote the divergence and Laplacian in $\mathbf{y}_t$, respectively (subscripts are omitted for clarity). We seek a diffusion process, run in reverse time, whose marginals coincide with $p(\cdot)$. Consider a reverse-time SDE with the same diffusion coefficient $g(\cdot)$ and some drift $\bar{f}(\cdot)$,

$$d\mathbf{Y}_t = \bar{f}(\mathbf{Y}_t, t) dt + g(t) d\bar{\mathbf{W}}_t, \quad (\text{A.8})$$

where $\bar{\mathbf{W}}_t$ is a Brownian motion in reverse time and dt is a negative increment. Its marginals $q(\cdot)$ satisfy the Fokker–Planck (reverse) equation (Anderson, 1982),

$$\partial_t q(\mathbf{y}_t) = -\nabla \cdot [\bar{f}(\mathbf{y}_t, t) q(\mathbf{y}_t)] - \frac{1}{2} g(t)^2 \nabla^2 q(\mathbf{y}_t). \quad (\text{A.9})$$

For the reverse process to reproduce the required marginals we require $q(\cdot) \equiv p(\cdot)$ for all t . Rewriting the diffusion term of (A.7) using $\nabla^2 p(\cdot) = \nabla \cdot (\nabla p(\cdot)) = \nabla \cdot (p(\cdot) \nabla \log p(\cdot))$, the forward equation (A.7) becomes

$$\partial_t p(\mathbf{y}_t) = -\nabla \cdot \left[(f(\mathbf{y}_t, t) - g(t)^2 \nabla \log p(\mathbf{y}_t)) p(\mathbf{y}_t) \right] - \frac{1}{2} g^2(t) \nabla^2 p(\mathbf{y}_t). \quad (\text{A.10})$$

Comparing (A.10) with (A.9) and setting $q(\cdot) = p(\cdot)$ identifies the reverse drift,

$$\bar{f}(\mathbf{y}_t, t) = f(\mathbf{y}_t, t) - g(t)^2 \nabla \log p(\mathbf{y}_t). \quad (\text{A.11})$$

Substituting $f(\mathbf{y}_t, t) = -\frac{1}{2}\beta_t \mathbf{y}_t$ and $g(t)^2 = \beta_t$ into (A.11) and (A.8) gives (4).

A.3 Denoising score-matching objective (6)

Here we give the derivation for the loss (6) used to train a neural network to approximate the score $\nabla \log p(\mathbf{Y}_t)$; the derivation for the loss (7) is similar in many respects, and not shown for brevity.

The starting point of the derivation is the explicit score-matching loss

$$J_{\text{ESM}}(\phi) = \mathbb{E}_{\mathbf{Y}_t} \left[\left\| \mathbf{s}_\phi(\mathbf{Y}_t, t) - \nabla \log p(\mathbf{Y}_t) \right\|^2 \right], \quad (\text{A.12})$$

which is minimised by the true score $\mathbf{s}_\phi(\mathbf{Y}_t, t) = \nabla \log p(\mathbf{Y}_t)$ for all t (as in Appendix A.2, here we omit subscripts on the gradient operator for clarity). Equation (A.12) however cannot be evaluated directly because $\nabla \log p(\mathbf{Y}_t)$ is not known in general. Following Vincent

(2011), we show that it is equivalent, up to a constant independent of ϕ , to the denoising score-matching loss

$$J_{\text{DSM}}(\phi) = \mathbb{E}_{\mathbf{Y}_0} \mathbb{E}_{\mathbf{Y}_t | \mathbf{Y}_0} \left[\|\mathbf{s}_\phi(\mathbf{Y}_t, t) - \nabla \log p(\mathbf{Y}_t | \mathbf{Y}_0)\|^2 \right], \quad (\text{A.13})$$

for all t , in which the conditional score $\nabla \log p(\mathbf{Y}_t | \mathbf{Y}_0)$ is available in closed form. Expanding the squared norms in (A.12) and (A.13), the $\|\mathbf{s}_\phi(\mathbf{Y}_t, t)\|^2$ terms are identical and the terms not involving $\mathbf{s}_\phi(\mathbf{Y}_t, t)$ are constant in ϕ ; the two objectives therefore differ only through their cross terms. Since $p(\mathbf{y}_t) \nabla \log p(\mathbf{y}_t) = \nabla p(\mathbf{y}_t)$, the cross term of (A.12) is

$$\mathbb{E}_{\mathbf{Y}_t} [\mathbf{s}_\phi(\mathbf{Y}_t, t)^\top \nabla \log p(\mathbf{Y}_t)] = \int \mathbf{s}_\phi(\mathbf{y}_t, t)^\top \nabla p(\mathbf{y}_t) d\mathbf{y}_t. \quad (\text{A.14})$$

Now, since $p(\mathbf{y}_t) = \int p(\mathbf{y}_0) p(\mathbf{y}_t | \mathbf{y}_0) d\mathbf{y}_0$, we have that

$$\nabla p(\mathbf{y}_t) = \int p(\mathbf{y}_0) \nabla p(\mathbf{y}_t | \mathbf{y}_0) d\mathbf{y}_0 = \int p(\mathbf{y}_0) p(\mathbf{y}_t | \mathbf{y}_0) \nabla \log p(\mathbf{y}_t | \mathbf{y}_0) d\mathbf{y}_0. \quad (\text{A.15})$$

Substituting (A.15) into (A.14) yields

$$\mathbb{E}_{\mathbf{Y}_t} [\mathbf{s}_\phi(\mathbf{Y}_t, t)^\top \nabla \log p(\mathbf{Y}_t)] = \mathbb{E}_{\mathbf{Y}_0} \mathbb{E}_{\mathbf{Y}_t | \mathbf{Y}_0} [\mathbf{s}_\phi(\mathbf{Y}_t, t)^\top \nabla \log p(\mathbf{Y}_t | \mathbf{Y}_0)].$$

Hence $J_{\text{ESM}}(\phi) = J_{\text{DSM}}(\phi) + \text{const}$, and the two losses share the same minimiser for each t (Vincent, 2011).

To find an optimal ϕ for all t , it is sufficient to minimise $J_{\text{DSM}}(\phi)$ uniformly over t . Integrating (A.13) with respect to time gives the objective $\mathcal{L}_S(\phi)$ in (6).

B Π GDM approximations for non-linear data models

This appendix details the approximations used to evaluate the noised likelihood (15) for Π GDM with the count, right-censoring and area-exceedance data models of Section 3.3.

B.1 Poisson data

Under the Poisson data model (18) the noised likelihood (15) is not available in closed form. To make the integral tractable, since $\mathbf{H}$ is an incidence matrix we approximate each factor $p(Z_i | Y_{0,i})$ by a Gaussian in $Y_{0,i}$. A second-order Taylor expansion of $\log p(Z_i | Y_{0,i})$ about the Tweedie estimate $\hat{Y}_{0,i}(\mathbf{Y}_t)$, with $u_i \equiv Y_{0,i} - \hat{Y}_{0,i}(\mathbf{Y}_t)$, gives

$$\log p(Z_i | Y_{0,i}) \approx \log p(Z_i | \hat{Y}_{0,i}(\mathbf{Y}_t)) + g_i u_i - \frac{1}{2} \Lambda_{ii} u_i^2,$$

where

$$g_i \equiv \left. \frac{\partial \log p(Z_i | Y_{0,i})}{\partial Y_{0,i}} \right|_{Y_{0,i} = \hat{Y}_{0,i}(\mathbf{Y}_t)}, \quad \Lambda_{ii} \equiv - \left. \frac{\partial^2 \log p(Z_i | Y_{0,i})}{\partial Y_{0,i}^2} \right|_{Y_{0,i} = \hat{Y}_{0,i}(\mathbf{Y}_t)},$$

are the log-likelihood first and second derivatives with respect to $Y_{0,i}$, respectively. Completing the square in u_i ,

$$g_i u_i - \frac{1}{2} \Lambda_{ii} u_i^2 = -\frac{\Lambda_{ii}}{2} (u_i - \Lambda_{ii}^{-1} g_i)^2 + \frac{g_i^2}{2\Lambda_{ii}},$$

whose final term is constant in $Y_{0,i}$. Introducing the pseudo-response

$$\tilde{Z}_i \equiv \hat{Y}_{0,i}(\mathbf{Y}_t) + \Lambda_{ii}^{-1} g_i,$$

gives $p(Z_i | Y_{0,i}) \propto \exp(-\frac{\Lambda_{ii}}{2}(Y_{0,i} - \tilde{Z}_i)^2)$. This is, up to a factor independent of $Y_{0,i}$, the density of a Gaussian pseudo-response

$$\tilde{Z}_i | Y_{0,i} \sim \text{Gau}(Y_{0,i}, \Lambda_{ii}^{-1}).$$

Stacking the m observed cells, the pseudo-responses satisfy $\tilde{\mathbf{Z}} = \mathbf{H}\mathbf{Y}_0 + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim \text{Gau}(\mathbf{0}, \mathbf{\Lambda}^{-1})$ and $\mathbf{\Lambda}$ the diagonal matrix comprising the $\Lambda_{ii}, i = 1, \dots, m$. The noised likelihood therefore follows from Section 3.2:

$$\tilde{\mathbf{Z}} | \mathbf{Y}_t \approx \text{Gau}(\mathbf{H}\hat{\mathbf{Y}}_0(\mathbf{Y}_t), \mathbf{H}\mathbf{V}_t\mathbf{H}^\top + \mathbf{\Lambda}^{-1}), \quad (\text{B.1})$$

and, as in the linear-Gaussian case, the guidance gradient follows by differentiating the resulting log-density through $\hat{\mathbf{Y}}_0(\mathbf{Y}_t)$.

B.2 Right censoring

Recall that under the right-censoring data model,

$$\log p(\mathbf{Z}, \mathbf{M} | \mathbf{Y}_0) = \sum_{i:M_i=0} \left[-\frac{1}{2} \left(\frac{Z_i - Y_{0,i}}{\sigma_\epsilon} \right)^2 \right] + \sum_{i:M_i=1} \log \Phi \left(\frac{Y_{0,i} - c}{\sigma_\epsilon} \right), \quad (\text{B.2})$$

up to an additive constant independent of $\mathbf{Y}_0$. Since the data are pixel-referenced and are conditionally independent given the latent process, we can write $p(\mathbf{Z}, \mathbf{M} | \mathbf{Y}_0) = \prod_{i=1}^n p_i(Z_i, M_i | Y_{0,i})$ where the factors are defined from (B.2). Now, assume that we can write $\mathbf{V}_t = v_t \mathbf{I}_n$ (see the discussion below (14)); then the integral in (15) is a product of one-dimensional integrals,

$$p(\mathbf{Z}, \mathbf{M} | \mathbf{Y}_t) \approx \prod_{i=1}^n \int p_i(Z_i, M_i | Y_{0,i}) \text{Gau}(Y_{0,i}; \hat{Y}_{0,i}(\mathbf{Y}_t), v_t) dY_{0,i}.$$

For an uncensored cell ($M_i = 0$), the noised likelihood component is

$$\int \text{Gau}(Z_i; Y_{0,i}, \sigma_\epsilon^2) \text{Gau}(Y_{0,i}; \hat{Y}_{0,i}(\mathbf{Y}_t), v_t) dY_{0,i} = \text{Gau}(Z_i; \hat{Y}_{0,i}(\mathbf{Y}_t), \sigma_\epsilon^2 + v_t), \quad i : M_i = 0.$$

For a censored cell ($M_i = 1$), note that $\Phi((Y_{0,i} - c)/\sigma_\epsilon) = \Pr(\epsilon_i \leq Y_{0,i} - c | Y_{0,i})$. Averaging over $Y_{0,i} \sim \text{Gau}(\hat{Y}_{0,i}(\mathbf{Y}_t), v_t)$ and applying the law of total probability,

$$\int \Phi \left(\frac{Y_{0,i} - c}{\sigma_\epsilon} \right) \text{Gau}(Y_{0,i}; \hat{Y}_{0,i}(\mathbf{Y}_t), v_t) dY_{0,i} = \Pr(\epsilon_i \leq Y_{0,i} - c) = \Pr(Y_{0,i} - \epsilon_i \geq c).$$

As $Y_{0,i}$ and ϵ_i are independent and Gaussian, $Y_{0,i} - \epsilon_i \sim \text{Gau}(\hat{Y}_{0,i}(\mathbf{Y}_t), \sigma_\epsilon^2 + v_t)$, and hence

$$\Pr(Y_{0,i} - \epsilon_i \geq c) = \Phi\left(\frac{\hat{Y}_{0,i}(\mathbf{Y}_t) - c}{\sqrt{\sigma_\epsilon^2 + v_t}}\right).$$

This leads to the noised likelihood component

$$\int \Phi\left(\frac{Y_{0,i} - c}{\sigma_\epsilon}\right) \text{Gau}(Y_{0,i}; \hat{Y}_{0,i}(\mathbf{Y}_t), v_t) dY_{0,i} = \Phi\left(\frac{\hat{Y}_{0,i}(\mathbf{Y}_t) - c}{\sqrt{\sigma_\epsilon^2 + v_t}}\right), \quad i : M_i = 1.$$

Taking logs of both components and evaluating at $\hat{\mathbf{Y}}_0(\mathbf{Y}_t)$ we obtain the noised log-likelihood

$$\log p(\mathbf{Z}, \mathbf{M} \mid \mathbf{Y}_t) \approx \sum_{i: M_i=0} \left[-\frac{1}{2} \left(\frac{Z_i - \hat{Y}_{0,i}(\mathbf{Y}_t)}{\sqrt{\sigma_\epsilon^2 + v_t}} \right)^2 \right] + \sum_{i: M_i=1} \log \Phi\left(\frac{\hat{Y}_{0,i}(\mathbf{Y}_t) - c}{\sqrt{\sigma_\epsilon^2 + v_t}}\right), \quad (\text{B.3})$$

up to an additive constant. Comparing (B.3) to (B.2) we see that conditioning on $\mathbf{Y}_t$ instead of $\mathbf{Y}_0$ effectively inflates the variance, as expected.

B.3 Area-exceedance functional

To incorporate the area-exceedance functional with ΠGDM , we first construct a Gaussian approximation of the data model. Define the normalised discrete weights

$$w_k(\mathbf{Y}_0) = \frac{\exp(-(k - h(\mathbf{Y}_0))^2 / (2\xi^2))}{\sum_{j=0}^n \exp(-(j - h(\mathbf{Y}_0))^2 / (2\xi^2))} \quad k \in \{0, \dots, n\},$$

and set

$$\mu(\mathbf{Y}_0) \equiv \mathbb{E}[Z \mid \mathbf{Y}_0] = \sum_{k=0}^n k w_k(\mathbf{Y}_0), \quad \gamma(\mathbf{Y}_0) \equiv \text{Var}(Z \mid \mathbf{Y}_0) = \sum_{k=0}^n k^2 w_k(\mathbf{Y}_0) - \mu(\mathbf{Y}_0)^2.$$

Then $p(Z \mid \mathbf{Y}_0) \approx \text{Gau}(\mu(\mathbf{Y}_0), \gamma(\mathbf{Y}_0))$. Both $\mu(\cdot)$ and $\gamma(\cdot)$ depend on $\mathbf{Y}_0$ through the ‘‘soft count’’ $h(\mathbf{Y}_0) = \sum_{i=1}^n \sigma_\kappa(Y_{0,i})$ where, recall, $\sigma_\kappa(Y_{0,i}) \equiv \text{expit}(\kappa(Y_{0,i} - c))$. Because ξ^2 is small, this discrete Gaussian concentrates almost all of its mass within one unit of $h(\mathbf{Y}_0)$. We therefore approximate $\gamma(\mathbf{Y}_0) \approx \gamma(\hat{\mathbf{Y}}_0(\mathbf{Y}_t))$, which is correspondingly a small variance controlled by ξ^2 . Under this approximation for the data model, (15) becomes

$$p(Z \mid \mathbf{Y}_t) \approx \int \text{Gau}(Z; \mu(\mathbf{Y}_0), \gamma(\hat{\mathbf{Y}}_0(\mathbf{Y}_t))) \text{Gau}(\mathbf{Y}_0; \hat{\mathbf{Y}}_0(\mathbf{Y}_t), \mathbf{V}_t) d\mathbf{Y}_0,$$

which is intractable since $\mu(\mathbf{Y}_0)$ depends non-linearly on $\mathbf{Y}_0$. We recover tractability by linearising the $\mu(\cdot)$ about $\hat{\mathbf{Y}}_0(\mathbf{Y}_t)$. Specifically, let

$$J_i(\mathbf{Y}_0) \equiv \partial \mu(\mathbf{Y}_0) / \partial Y_{0,i} = (d\mu(\mathbf{Y}_0) / dh(\mathbf{Y}_0)) \partial h(\mathbf{Y}_0) / \partial Y_{0,i}.$$

Differentiating the weights w_k with respect to h gives $\partial w_k / \partial h = w_k (k - \mu) / \xi^2$, so that

$$\frac{d\mu}{dh} = \sum_{k=0}^n k \frac{\partial w_k}{\partial h} = \frac{1}{\xi^2} \left(\sum_{k=0}^n k^2 w_k - \mu(\mathbf{Y}_0)^2 \right) = \frac{\gamma(\mathbf{Y}_0)}{\xi^2},$$

Combining this with $\partial h(\mathbf{Y}_0) / \partial Y_{0,i} = \kappa \sigma_\kappa(Y_{0,i}) (1 - \sigma_\kappa(Y_{0,i}))$ yields

$$J_i(\mathbf{Y}_0) = \frac{\gamma(\mathbf{Y}_0)}{\xi^2} \kappa \sigma_\kappa(Y_{0,i}) (1 - \sigma_\kappa(Y_{0,i})). \quad (\text{B.4})$$

A first-order Taylor expansion of the mean is then given by,

$$\mu(\mathbf{Y}_0) \approx \mu(\hat{\mathbf{Y}}_0(\mathbf{Y}_t)) + \mathbf{J}^\top(\hat{\mathbf{Y}}_0(\mathbf{Y}_t)) (\mathbf{Y}_0 - \hat{\mathbf{Y}}_0(\mathbf{Y}_t)),$$

where $\mathbf{J}(\cdot) \equiv (J_1(\cdot), \dots, J_n(\cdot))^\top$. We can now apply the “re-parameterisation trick” to write

$$Z = \mu(\hat{\mathbf{Y}}_0(\mathbf{Y}_t)) + \mathbf{J}(\hat{\mathbf{Y}}_0(\mathbf{Y}_t))^\top (\mathbf{Y}_0 - \hat{\mathbf{Y}}_0(\mathbf{Y}_t)) + \eta, \quad \eta \sim \text{Gau}(0, \gamma(\hat{\mathbf{Y}}_0(\mathbf{Y}_t))),$$

so that, when substituting in for $\mathbf{Y}_0 \sim \text{Gau}(\hat{\mathbf{Y}}_0(\mathbf{Y}_t), \mathbf{V}_t)$, we have that

$$Z \mid \mathbf{Y}_t \approx \text{Gau}\left(\mu(\hat{\mathbf{Y}}_0(\mathbf{Y}_t)), \tilde{v}_t\right), \quad \tilde{v}_t \equiv \mathbf{J}(\hat{\mathbf{Y}}_0)^\top \mathbf{V}_t \mathbf{J}(\hat{\mathbf{Y}}_0) + \gamma(\hat{\mathbf{Y}}_0(\mathbf{Y}_t)). \quad (\text{B.5})$$

Supplementary Material

S.1 Computational details

For the forward process (2) we use a variance-preserving schedule with linear $\beta_t = \beta_0 + (\beta_1 - \beta_0)t$, $\beta_0 = 0.1$, $\beta_1 = 20$, and $T = 1$, so that $\alpha_T \approx 0.007$ and hence $\mathbf{Y}_T \sim \text{Gau}(\mathbf{0}, \mathbf{I}_n)$.

To numerically integrate when reverse sampling, we discretise (4) on a grid $T = t_T > t_{T-1} > \dots > t_0 = 0$ and, on this grid, we use an update of the form

$$\mathbf{Y}_{t_{k-1}} = \mu_\phi(\mathbf{Y}_{t_k}, t_k) + \tilde{\sigma}_{t_k} \boldsymbol{\omega}_{t_k}, \quad \boldsymbol{\omega}_{t_k} \sim \text{Gau}(\mathbf{0}, \mathbf{I}_n), \quad (\text{S.1})$$

for $k = T, \dots, 1$, where $\mu_\phi(\mathbf{Y}_{t_k}, t_k)$ corresponds to the reverse-time dynamics in (4), evaluated using $\mathbf{r}_\phi(\mathbf{Y}_{t_k}, t_k)$, and $\tilde{\sigma}_{t_k}$ is a step-dependent scaling specified by the diffusion schedule. We construct (S.1) using the Euler–Maruyama scheme of Song et al. (2021) as implemented by Moss et al. (2026), on a time grid $\{t_k\}$ spaced uniformly in the log signal-to-noise ratio $\log(\alpha_t^2/\sigma_t^2)$.

At every reverse step we soft-norm-clip the guidance contribution to the reverse drift via a tanh squashing saturating at 100. For the first Gaussian process example in Section 4.1 the norm stayed below the bound for both DPS and IIGDM. For DPS in the other experiments the raw guidance norm grew by several orders of magnitude near $t = 0$, and clipping was necessary. Guidance with IIGDM is better conditioned and rarely required clipping.

In Section 4.1 the Gaussian-process score is available in closed form, so no neural network is needed. For the non-Gaussian process models, as score network we used a U-Net with circular padding for the periodic Ginzburg–Landau domain (Section 4.2) and zero padding for the bounded methane domain (Section 5). The U-Nets were trained using (7) with Adam (learning rate 2×10^{-4} , batch size 128, exponential moving average with decay 0.999). For more details on the neural networks, please see the supplementary code.

A convolutional score network requires a full rectangular array, whereas the methane flux field of Section 5 is defined only on the $n = 122$ land cells. We therefore embed each simulated flux field in the enclosing 20×20 longitude–latitude array, and fill the remaining 278 (sea and out-of-domain) cells with the median flux of the simulated ensemble; the higher-resolution configurations of Section 5.3 are treated identically. This choice does not affect the inferences: the transport operator $\mathbf{H}$ in (22) has zero columns at the inactive cells, and every predictive sample is restricted to the land cells before any diagnostic is computed.

S.2 Comparison with neural conditional simulation

The guided-diffusion framework defers the choice of data model to prediction time, whereas NCS (Walchessen et al., 2025) requires the data model to be fixed before its conditional score network $\tilde{\mathbf{r}}_\phi(\mathbf{Y}_t, \mathbf{Z}, t)$ is trained (Section 2). To illustrate the practical consequence of this difference, we conduct an experiment on the Ginzburg–Landau amplitude field of Section 4.2, using the same held-out realisations and the same training fields as the guided approaches, so that the two approaches differ only in how the conditioning information is

incorporated: NCS learns a single conditional score network, whereas guided diffusion pairs the unconditional process-model score with the data-model likelihood at prediction time.

We train an NCS network on a missing-at-random (MAR) Gaussian data model. For each simulated field, a random proportion of the grid cells are retained (where the observed proportion is uniformly generated between 0.02 and 0.10). Gaussian measurement error of known standard deviation $\sigma_\epsilon = 0.1$ is then added, and the network is trained on the corresponding conditional score-matching objective (8). We then apply the trained network in two settings. In the first setting the data are generated with missingness at random, with an observed proportion of 5%. In the second setting, the data instead arise from the upper-tail censoring model of Section 4.2, in which a cell is observed only when its noisy measurement $Z_i < c$, $c = 1$. In both settings we compare NCS against guided diffusion (DPS and IIGDM) with the appropriate data-model, and we report RMSE, 90% coverage, ES, and CRPS over 50 realisations. For the censoring setting we report these diagnostics over the censored cells.

Table S.1 summarises the results, and Figure S.1 shows predictions for one realisation. Under the matched MAR model, NCS is similar to guided diffusion. However, under the censoring model, the predictive accuracy of NCS deteriorates, as expected. The reason for this deterioration is that the NCS network is trained for data that are missing at random with a missingness proportion of between 0.90 and 0.98, and that have a measurement error standard deviation of $\sigma_\epsilon = 0.1$. The guided samplers, on the other hand, exploit the correct censoring likelihood (19) without the need for any retraining. Therefore, NCS performs well and arguably should be used when the data model is known before training time, but cannot accommodate a change in the data model without retraining. This is a limitation that the guided-diffusion framework removes.

Table S.1: Comparison of neural conditional simulation (NCS) with guided diffusion (DPS and IIGDM) on the Ginzburg–Landau amplitude field. The NCS network is trained on a missing-at-random (MAR) Gaussian data model and applied, without retraining, both to MAR data and to the upper-tail censoring model of Section 4.2. RMSE (of the predictive mean), 90%-interval coverage, ES, and CRPS are computed against the true field, as averages over 50 realisations; for the censoring setting the diagnostics are computed over the censored cells.

Scenario	Method	RMSE	Cov. (0.90)	ES	CRPS
MAR	NCS	0.154	0.893	3.47	0.079
	DPS	0.156	0.856	3.54	0.081
	IIGDM	0.158	0.903	3.57	0.081
Censoring	NCS	0.040	0.739	0.39	0.022
	DPS	0.010	0.816	0.10	0.006
	IIGDM	0.011	0.860	0.11	0.006

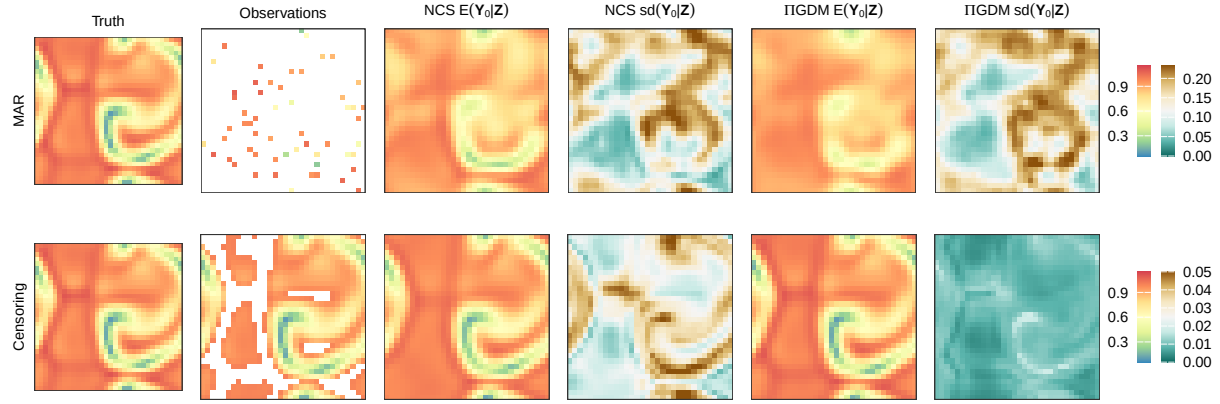


Figure S.1: Predictions of a Ginzburg–Landau amplitude field for one realisation under the MAR model (top) and the upper-tail censoring model (bottom). Columns show the true field, the observations (missing or censored cells in white), and the predictive mean and standard deviation for NCS and for IIGDM. The quantitative differences between the methods over several realisations are reported in Table S.1.

S.3 Additional Figures

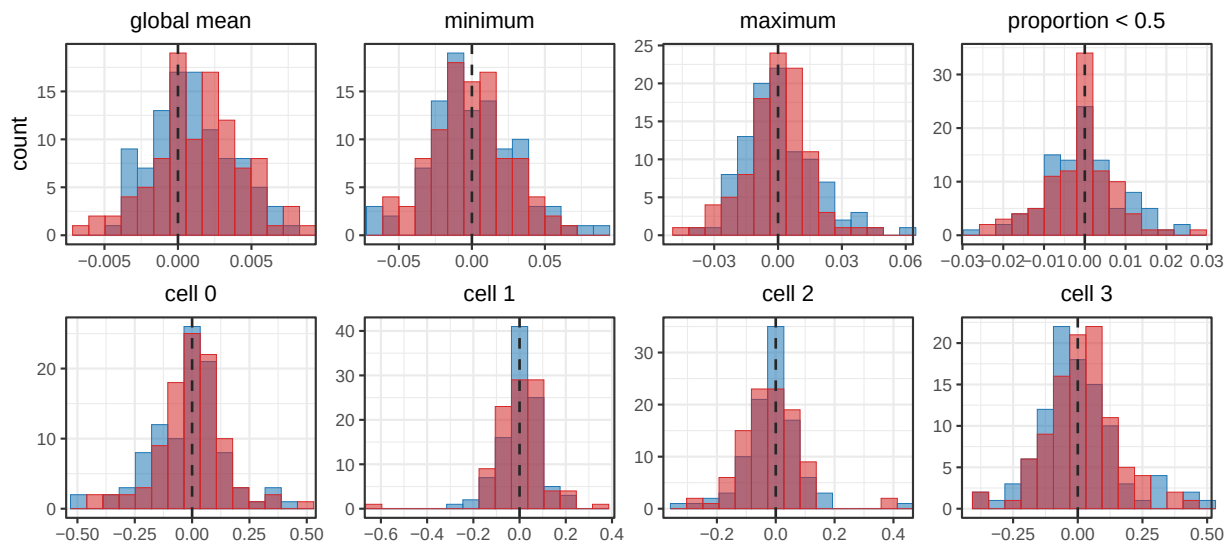


Figure S.2: Functional residuals for the areal aggregation experiment of Section 4.2 (100 independent realisations) for DPS (blue) and IIGDM (red). The dashed line is at zero.

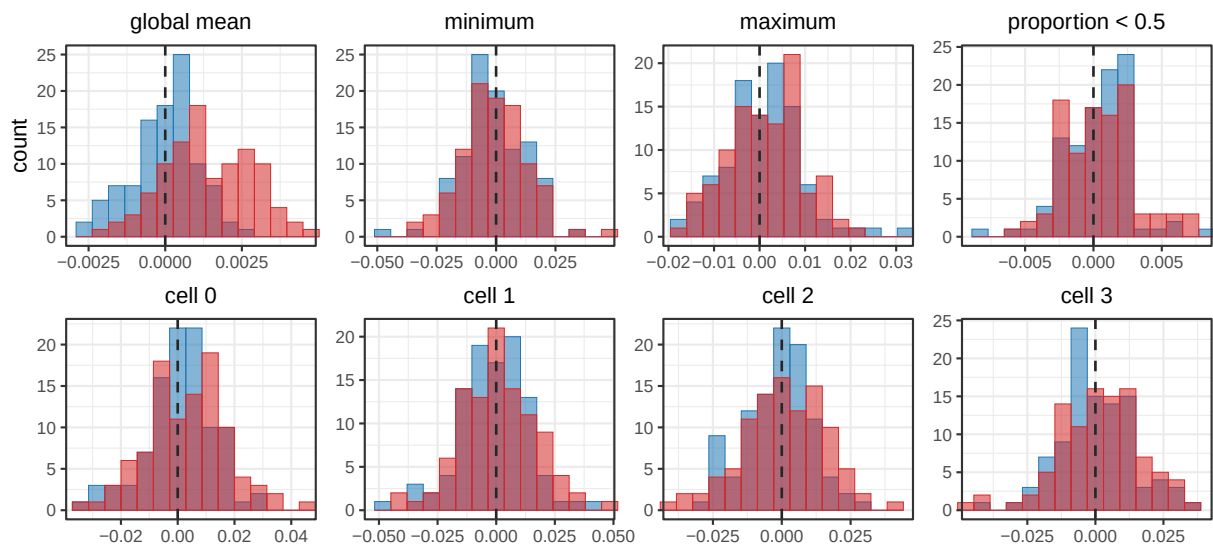


Figure S.3: Same as Figure S.2 but for the censored data experiment.

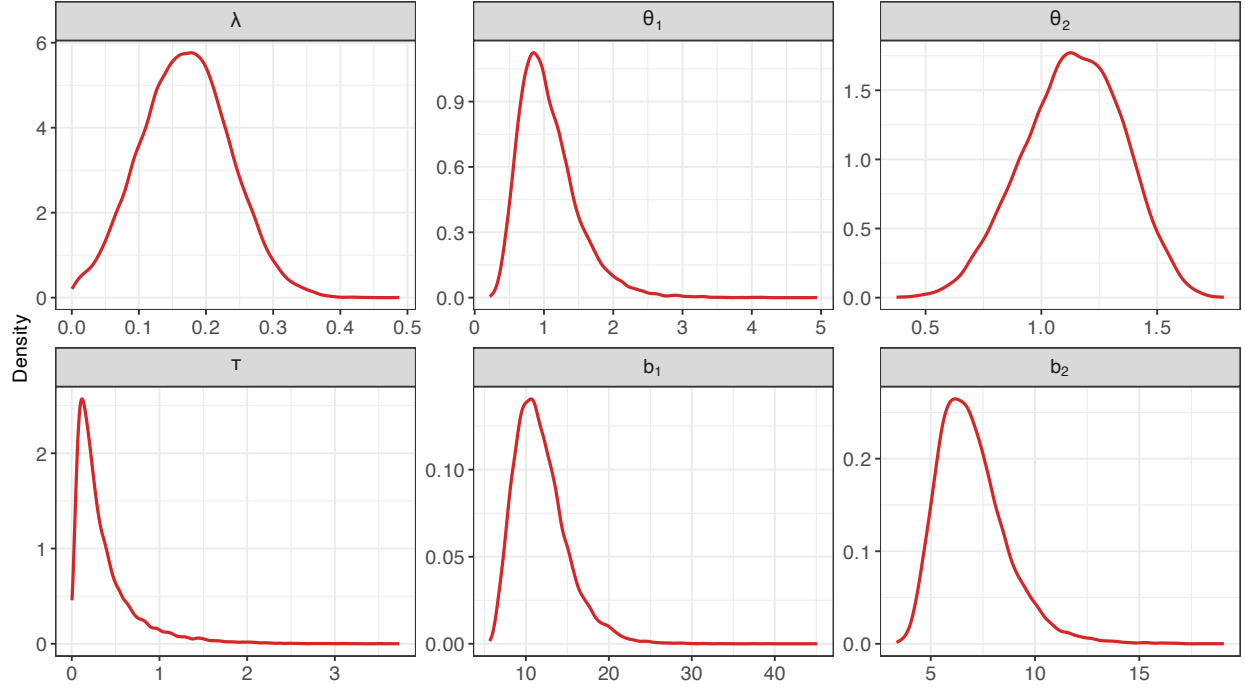


Figure S.4: Prior distributions used in the methane flux experiment (26), elicited from the inventory alone. Panels show the induced model-scale parameters $(\lambda, \theta_1, \theta_2, \tau, b_1, b_2)$ obtained from independent informative priors on the natural-scale parameters $(\lambda, \log \theta_1, \text{logit}(\theta_2/2), \log \tilde{b}_1, \log \tilde{b}_2, \log c_V)$ via (25).

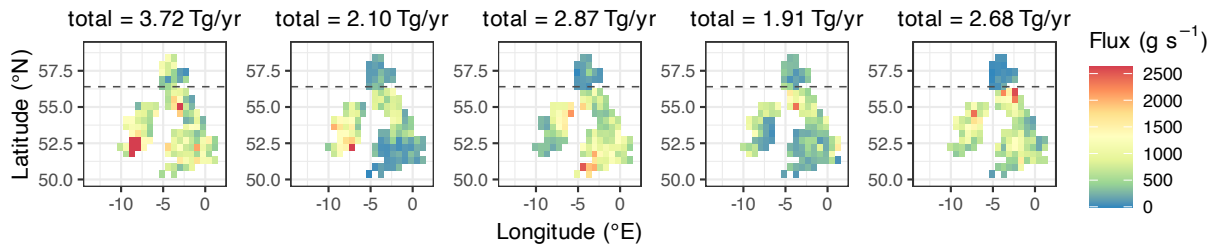


Figure S.5: Five unconditional draws from $p(\mathbf{Y}_0)$ via reverse diffusion, each annotated with the domain flux total. For context, the methane inventory flux is shown in Figure 7 and has a total of 2.73 Tg yr^{-1} .

Table S.2: Region-level total methane flux (g s^{-1}): true (inventory) value, and predictive mean $\pm$ predictive standard deviation for the two guided samplers (DPS and Π GDM) and MCMC (Section 5.2).

Territory	Truth	DPS	Π GDM	MCMC
England	45,604	$45,699 \pm 1,333$	$45,589 \pm 1,491$	$45,184 \pm 1,614$
Scotland	10,682	$13,391 \pm 1,561$	$13,780 \pm 1,809$	$13,248 \pm 1,713$
Wales	7,888	$8,246 \pm 606$	$8,059 \pm 671$	$8,102 \pm 739$
Northern Ireland	6,318	$5,484 \pm 861$	$5,411 \pm 935$	$5,578 \pm 956$
Ireland	16,097	$14,499 \pm 1,163$	$14,529 \pm 1,356$	$14,533 \pm 1,424$

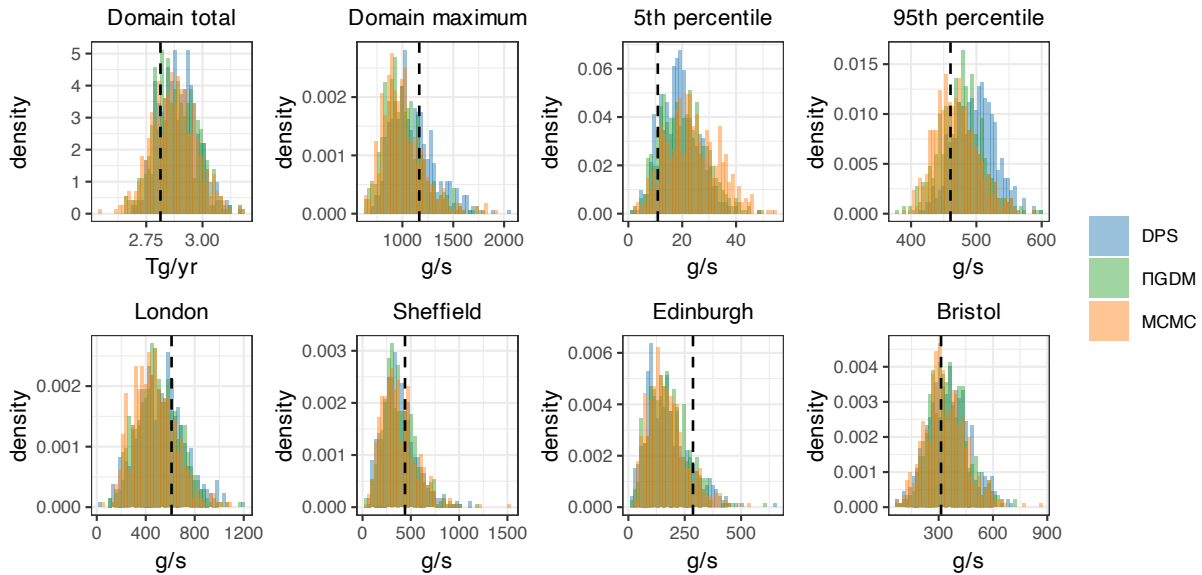


Figure S.6: As Figure 10 but for $n = 487$: predictive distributions of eight functionals of the flux field for DPS, Π GDM, and MCMC.

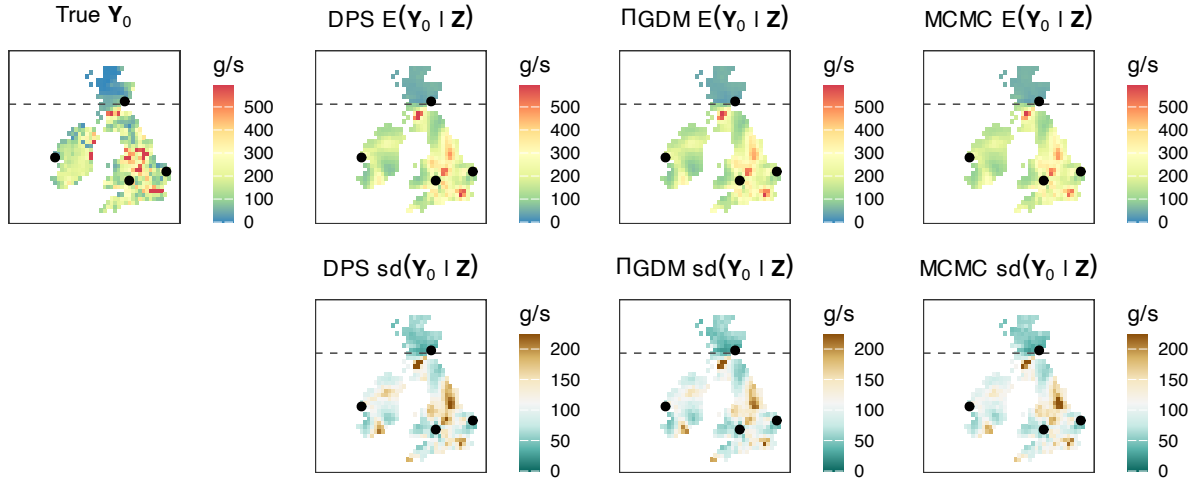


Figure S.7: As Figure 7 but for $n = 487$: per-cell predictive mean and predictive standard deviation for DPS, Π GDM, and MCMC.

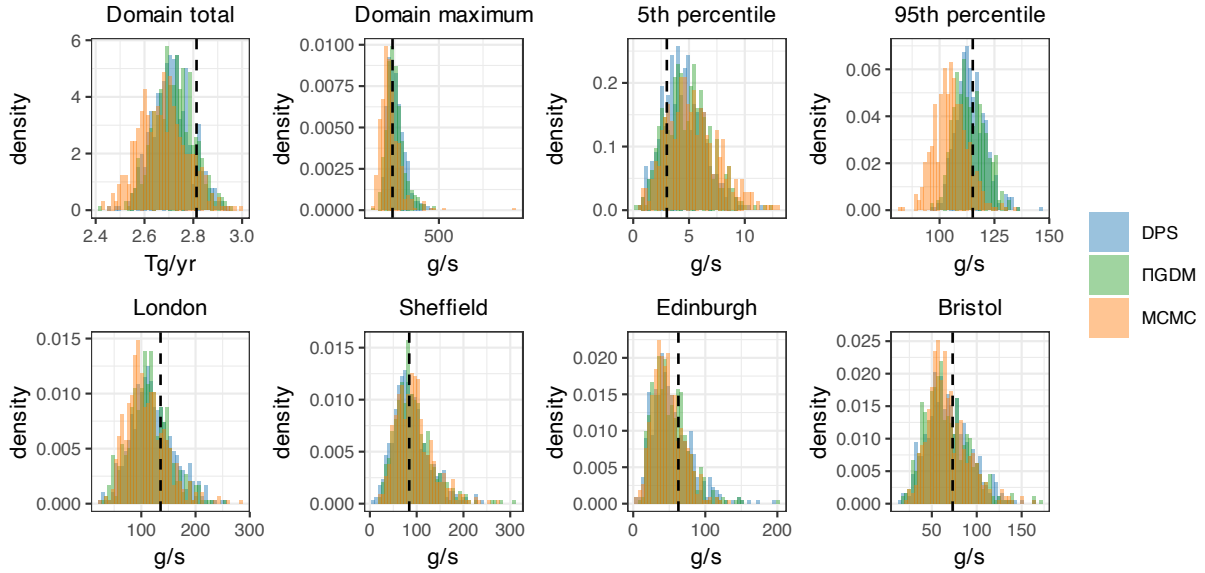


Figure S.8: As Figure 10 but for $n = 1948$: predictive distributions of eight functionals of the flux field for DPS, Π GDM, and MCMC.

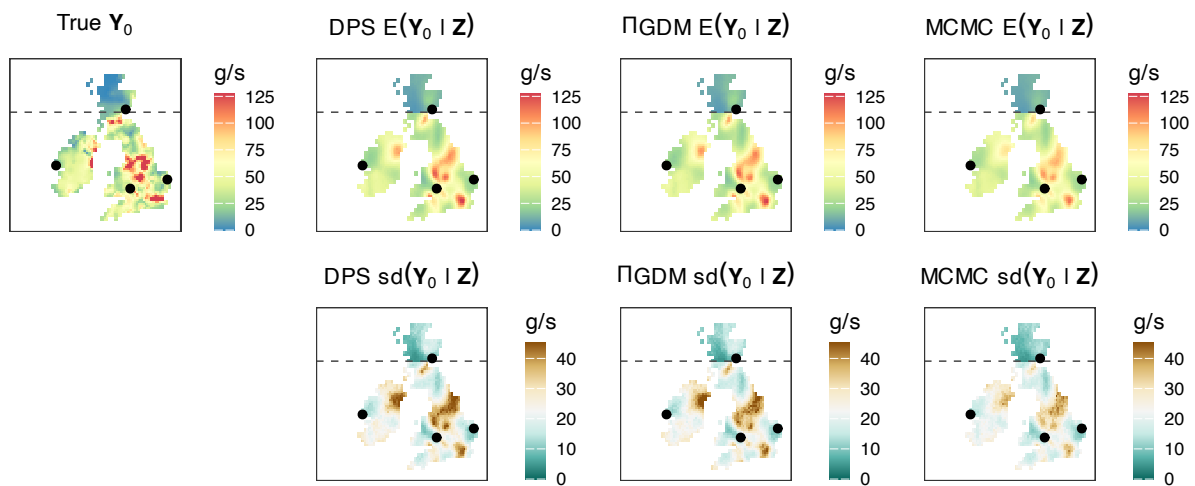


Figure S.9: As Figure 7 but for $n = 1948$: per-cell predictive mean and predictive standard deviation for DPS, Π GDM, and MCMC.