Supplementary to:
Geometric Neural Diffusion Processes

A Organisation of appendices

In this supplementary, we first introduce in App. B an Ornstein Uhlenbeck process on function space (via finite marginals) along with several score approximations. Then in App. C, we show how this methodology extend to manifold-valued inputs or outputs. Later in App. D, we derive sufficient conditions for this introduced model to yield a group invariant process. What's more in App. E, we study some conditional sampling schemes. Eventually in App. F, we give a thorough description of experimental settings along with additional empirical results.

B Ornstein Uhlenbeck on function space

B.1 Multivariate Ornstein-Uhlenbeck process

First, we aim to show that we can define a stochastic process on an infinite dimensional function space, by defining the joint finite marginals $\mathbf{Y}(x)$ as the solution of a multidimensional Ornstein-Uhlenbeck process. In particular, for any set of input $x = (x_1, \cdots, x_k) \in \mathcal{X}^k$, we define the joint marginal as the solution of the following SDE

$$\mathrm{d}\tilde{\mathbf{Y}}_t(x) = (m(x) - \tilde{\mathbf{Y}}_t(x))/2 \; \beta_t \mathrm{d}t + \sqrt{\beta_t K(x,x)}\mathrm{d}\mathbf{B}_t \; . \tag{8}$$

Proposition B.1. *(Phillips et al., 2022) We assume we are given a data process $(\mathbf{Y}_0(x))_{x\in\mathcal{X}}$ and we denote by $\mathbf{G} \sim \mathrm{GP}(0, k)$ a Gaussian process with zero mean and covariance. Then let's define*

$$\mathbf{Y}_t \triangleq e^{-\frac{1}{2}\cdot\int_{s=0}^t \beta_s ds}\, \mathbf{Y}_0 + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^t \beta_s ds}\right) m + \left(1 - e^{-\int_{s=0}^t \beta_s ds}\right)^{1/2} \mathbf{G}.$$

Then $(\mathbf{Y}_t(x))_{x\in\mathcal{X}}$ is a stochastic process (by virtue of being a linear combination of stochastic processes). We thus have that $\mathbf{Y}_t \xrightarrow[t\to 0]{a.s.} \mathbf{Y}_0$ and $\mathbf{Y}_t \xrightarrow[t\to\infty]{a.s.} \mathbf{Y}_\infty$ with $\mathbf{Y}_\infty \sim \mathrm{GP}(m, k)$, so effectively $(\mathbf{Y}_t(x))_{t\in\mathbb{R}_+, x\in\mathcal{X}}$ interpolates between the data process and this limiting Gaussian process. Additionally, $\mathcal{L}(\mathbf{Y}_t|\mathbf{Y}_0 = y_0) = \mathrm{GP}(m_t, K_t)$ with $m_t = e^{-\frac{1}{2}\cdot\int_{s=0}^t \beta_s ds}\, y_0 + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^t \beta_s ds}\right) m$ and $\Sigma_t = \left(1 - e^{-\int_{s=0}^t \beta_s ds}\right) K$. Furthermore, $(\mathbf{Y}_t(x))_{t\in\mathbb{R}_+, x\in\mathcal{X}}$ is the solution of the SDE in (8).

Proof. We aim to compute the mean and covariance of the process $(\mathbf{Y}_t)_{t\geq 0}$ described by the SDE (3). First let's recall the time evolution of the mean and covariance of the solution from a multivariate Ornstein-Uhlenbeck process given by

$$\mathrm{d}\mathbf{Y}_t = f(\mathbf{Y}_t, t)\mathrm{d}t + L(\mathbf{Y}_t, t)\mathrm{d}\mathbf{B}_t. \tag{9}$$

We know that the time evolution of the mean and the covariance are given respectively by Särkkä and Solin (2019)

$$\frac{\mathrm{d}m_t}{\mathrm{d}t} = \mathrm{E}[f(\mathbf{Y}_t, t)] \tag{10}$$

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = \mathrm{E}[f(\mathbf{Y}_t, t)(m_t - \mathbf{Y}_t)^\top] + \mathrm{E}[(m_t - \mathbf{Y}_t)f(\mathbf{Y}_t, t)^\top] + \mathrm{E}[L(\mathbf{Y}_t, t)L(\mathbf{Y}_t, t)^\top]. \tag{11}$$

Plugging in the drift $f(\mathbf{Y}_t, t) = 1/2 \cdot (m - \mathbf{Y}_t)\beta_t$ and diffusion term $L(\mathbf{Y}_t, t) = \sqrt{\beta_t K}$ from (3), we get

$$\frac{\mathrm{d}m_t}{\mathrm{d}t} = 1/2 \cdot (m - \mathbf{Y}_t)\beta_t \tag{12}$$

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = \beta_t \left[K - \Sigma_t\right]. \tag{13}$$

Solving these two ODEs we get

$$m_t = e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds} m_0 + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\right) m \tag{14}$$

$$\Sigma_t = K + e^{-\int_{s=0}^{t}\beta_s ds}\left(\Sigma_0 - K\right) \tag{15}$$

with $m_0 \triangleq \mathrm{E}[\mathbf{Y}_0]$ and $\Sigma_0 \triangleq \mathrm{Cov}[\mathbf{Y}_0]$.

Now let's compute the first two moments of $(\mathbf{Y}_t(x))_{x\in\mathcal{X}}$. We have

$$\mathrm{E}[\mathbf{Y}_t] = \mathrm{E}\left[e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\,\mathbf{Y}_0 + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\right) m + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\right)\mathbf{G}\right] \tag{16}$$

$$= e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds} m_0 + \left(1 - e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\right) m \tag{17}$$

$$= m_t \tag{18}$$

$$\mathrm{Cov}[\mathbf{Y}_t] = \mathrm{Cov}\left[e^{-\frac{1}{2}\cdot\int_{s=0}^{t}\beta_s ds}\,\mathbf{Y}_0\right] + \mathrm{Cov}\left[\left(1 - e^{-\int_{s=0}^{t}\beta_s ds}\right)^{1/2}\mathbf{G}\right] \tag{19}$$

$$= e^{-\int_{s=0}^{t}\beta_s ds}\,\Sigma_0 + \left(1 - e^{-\int_{s=0}^{t}\beta_s ds}\right) K \tag{20}$$

$$= K + e^{-\int_{s=0}^{t}\beta_s ds}\left(\Sigma_0 - K\right) \tag{21}$$

$$= \Sigma_t\ . \tag{22}$$

□

B.2 Conditional score

Hence, condition on $\mathbf{Y}_0$ the score is the gradient of the log Gaussian characterised by mean $m_{t|0} = e^{-\frac{1}{2}B(t)}\mathbf{Y}_0$ and $\Sigma_{t|0} = (1 - e^{-B(t)})\mathrm{K}$ with $B(t) = \int_0^t \beta(s)ds$ which can be derived from the above marginal mean and covariance with $m_0 = \mathbf{Y}_0$ and $\Sigma_0 = 0$.

$$\nabla_{\mathbf{Y}_t}\log p_t(\mathbf{Y}_t|\mathbf{Y}_0) = \nabla_{\mathbf{Y}_t}\log\mathcal{N}\left(\mathbf{Y}_t|m_{t|0},\Sigma_{t|0}\right) \tag{23}$$

$$= \nabla_{\mathbf{Y}_t} - 1/2(\mathbf{Y}_t - m_{t|0})^\top\Sigma_{t|0}^{-1}(\mathbf{Y}_t - m_{t|0}) + c \tag{24}$$

$$= -\Sigma_{t|0}^{-1}(\mathbf{Y}_t - m_{t|0}) \tag{25}$$

$$= -\mathrm{L}_{t|0}^{-\top}\mathrm{L}_{t|0}^{-1}\mathrm{L}_{t|0}\epsilon \tag{26}$$

$$= -\mathrm{L}_{t|0}^{-\top}\epsilon \tag{27}$$

where $\mathrm{L}_{t|0}$ denotes the Cholesky decomposition of $\Sigma_{t|0} = \mathrm{L}_{t|0}\mathrm{L}_{t|0}^\top$, and $\mathbf{Y}_t = m_{t|0} + \mathrm{L}_{t|0}\epsilon$.

Then we can plugin our learnt (preconditioned) score into the backward SDE 4 which gives

$$\mathrm{d}\bar{\mathbf{Y}}_t|x = \left[-(m(x) - \bar{\mathbf{Y}}_t)/2 + \mathrm{K}(x,x)\nabla_{\bar{\mathbf{Y}}_t}\log p_{T-t}(t,x,\bar{\mathbf{Y}}_t)\right]\mathrm{d}t + \sqrt{\beta_t\mathrm{K}(x,x)}\beta_t\mathrm{d}\mathbf{B}_t \tag{28}$$

B.3 Several score parametrisations

In this section, we discuss several parametrisations of the neural network and the objective.

For the sake of versatility, we opt to employ the symbol D_θ for the network instead of s_θ as mentioned in the primary text, as it allows us to approximate not only the score but also other quantities from which the score can be derived. In full generality, we use a residual connection, weighted by $c_{\text{out}}, c_{\text{skip}} : \mathbb{R} \to \mathbb{R}$, to parameterise the network

$$D_\theta(t, \boldsymbol{Y}_t) = c_{\text{skip}}(t)\boldsymbol{Y}_t + c_{\text{out}}(t)F_\theta(t, \boldsymbol{Y}_t). \tag{29}$$

We recall that the input to the network is time t, and the noised vector $\boldsymbol{Y}_t = \boldsymbol{\mu}_{t|0} + \boldsymbol{n}$, where $\boldsymbol{\mu}_{t|0} = \mathrm{e}^{-B(t)/2}\boldsymbol{Y}_0$ and $\boldsymbol{n} \sim \mathcal{N}(0, \Sigma_{t|0})$ with $\Sigma_{t|0} = (1 - \mathrm{e}^{-B(t)})K$. The gram matrix K corresponds to $k(X, X)$ with k the limiting kernel. We denote by $\mathrm{L}_{t|0}$ and S respectively the Cholesky decomposition of $\Sigma_{t|0} = \mathrm{L}_{t|0}\mathrm{L}_{t|0}^\top$ and $\mathrm{K} = \mathrm{S}\mathrm{S}^\top$.

The denoising score matching loss weighted by $\Lambda(t)$ is given by

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) - \nabla_{\boldsymbol{Y}_t}\log p_t(\boldsymbol{Y}_t|\boldsymbol{Y}_0)\|^2_{\Lambda(t)}\right] \tag{30}$$

Table 3: Summary of different score parametrisations as well as the values for c_{skip} and c_{out} that we found to be optimal, based on the recommendation from Karras et al. (2022, Appendix B.6).

	No precond.	Precond. K	Precond. $S^\top$	Predict $\boldsymbol{Y}_0$
c_{skip}	0	0	0	1
c_{out}	$(\sigma_{t\|0} + 10^{-3})^{-1}$	$(\sigma_{t\|0} + 10^{-3})^{-1}$	$(\sigma_{t\|0} + 10^{-3})^{-1}$	1
Loss	$\|\sigma_{t\|0} S^\top D_\theta + \boldsymbol{z}\|_2^2$	$\|\sigma_{t\|0} D_\theta + S\boldsymbol{z}\|_2^2$	$\|\sigma_{t\|0} D_\theta + \boldsymbol{z}\|_2^2$	$\|D_\theta - \boldsymbol{Y}_0\|_2^2$
$K\nabla \log p_t$	KD_θ	D_θ	SD_θ	$-\Sigma_{t\|0}^{-1}(\boldsymbol{Y}_t - \mathrm{e}^{-\frac{B(t)}{2}} D_\theta)$

No preconditioning By reparametrisation, let $\boldsymbol{Y}_t = \boldsymbol{\mu}_{t|0} + L_{t|0}\boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the loss from Eq. (30) can be written as

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + \Sigma_{t|0}^{-1}(\boldsymbol{Y}_t - \boldsymbol{\mu}_{t|0})\|_{\Lambda(t)}^2\right] \tag{31}$$

$$= \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + \Sigma_{t|0}^{-1} L_{t|0}\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{32}$$

$$= \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + L_{t|0}^{-\top}\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{33}$$

$$\tag{34}$$

Choosing $\Lambda(t) = \Sigma_{t|0} = L_{t|0}L_{t|0}^\top$ we obtain

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|L_{t|0}^\top D_\theta(t, \boldsymbol{Y}_t) + \boldsymbol{z}\|_2^2\right] \tag{35}$$

$$= \mathbb{E}\left[\|\sigma_{t|0} S^\top D_\theta(t, \boldsymbol{Y}_t) + \boldsymbol{z}\|_2^2\right]. \tag{36}$$

Preconditioning by K Alternatively, one can train the neural network to approximate the preconditioned score $D_\theta \approx \mathbf{K}\nabla_{\boldsymbol{Y}_t} \log p_t(\boldsymbol{Y}_t|\boldsymbol{Y}_0)$. The loss, weighted by $\Lambda = \sigma_{t|0}^2\mathbf{I}$, is then given by

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + K\, L_{t|0}^{-\top}\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{37}$$

$$= \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + \sigma_{t|0}^{-1} S\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{38}$$

$$= \mathbb{E}\left[\|\sigma_{t|0} D_\theta(t, \boldsymbol{Y}_t) + S\boldsymbol{z}\|_2^2\right]. \tag{39}$$

Precondition by $S^\top$ A variation of the previous one, is to precondition the score by the transpose Cholesky of the limiting kernel gram matrix, such that $D_\theta \approx S^\top \nabla_{\boldsymbol{Y}_t} \log p_t(\boldsymbol{Y}_t|\boldsymbol{Y}_0)$.

The loss, weighted by $\Lambda = \sigma_{t|0}^2\mathbf{I}$, becomes

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + S^\top\, L_{t|0}^{-\top}\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{40}$$

$$= \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) + \sigma_{t|0}^{-1}\boldsymbol{z}\|_{\Lambda(t)}^2\right] \tag{41}$$

$$= \mathbb{E}\left[\|\sigma_{t|0} D_\theta(t, \boldsymbol{Y}_t) + \boldsymbol{z}\|_2^2\right]. \tag{42}$$

Predicting $\boldsymbol{Y}_0$ Finally, an alternative strategy is to predict $\boldsymbol{Y}_0$ from a noised version $\boldsymbol{Y}_t$. In this case, the loss takes the simple form

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|D_\theta(t, \boldsymbol{Y}_t) - \boldsymbol{Y}_0\|_2^2\right].$$

The score can be computed from the network's prediction following

$$\nabla \log p_t(\boldsymbol{Y}_t|\boldsymbol{Y}_0) = -\Sigma_{t|0}^{-1}(\boldsymbol{Y}_t - \boldsymbol{\mu}_{t|0}) \tag{43}$$

$$= -\Sigma_{t|0}^{-1}(\boldsymbol{Y}_t - \mathrm{e}^{-B(t)/2}\boldsymbol{Y}_0) \tag{44}$$

$$\approx -\Sigma_{t|0}^{-1}\left(\boldsymbol{Y}_t - \mathrm{e}^{-B(t)/2} D_\theta(t, \boldsymbol{Y}_t)\right) \tag{45}$$

$$\tag{46}$$

Table 3 summarises the different options for parametrising the score as well as the values for c_{skip} and c_{out} that we found to be optimal, based on the recommendation from Karras et al. (2022, Appendix B.6). In practice, we found the precondition by K parametrisation to produce the best results, but we refer to App. F.1.3 for a more in-depth ablation study.

B.4 Exact (marginal) score in Gaussian setting

Interpolating between Gaussian processes $GP(m_0, \Sigma_0)$ and $GP(m, \mathrm{K})$

$$\begin{aligned}
\mathrm{K}\nabla_{\bar{\mathbf{Y}}_t} \log p_t(\mathbf{Y}_t) &= -\mathrm{K}\Sigma_t^{-1}(\mathbf{Y}_t - m_t) && (47)\\
&= -\mathrm{K}[\mathrm{K} + e^{-\int_{s=0}^{t}\beta_s ds}(\Sigma_0 - \mathrm{K})]^{-1}(\mathbf{Y}_t - m_t) && (48)\\
&= -\mathrm{K}(\mathrm{L}_t {\mathrm{L}_t}^{\top})^{-1}(\mathbf{Y}_t - m_t) && (49)\\
&= -\mathrm{K}{\mathrm{L}_t}^{\top -1}{\mathrm{L}_t}^{-1}(\mathbf{Y}_t - m_t) && (50)\\
& && (51)
\end{aligned}$$

with $\Sigma_t = \mathrm{K} + e^{-\int_{s=0}^{t}\beta_s ds}(\Sigma_0 - \mathrm{K}) = \mathrm{L}_t {\mathrm{L}_t}^{\top}$ obtained via Cholesky decomposition.

B.5 Langevin dynamics

Under mild assumptions on $\nabla \log p_{T-t}$ (Durmus and Moulines, 2016) the following SDE

$$\mathrm{d}\mathbf{Y}_s = \tfrac{1}{2}\mathrm{K}\nabla \log p_{T-t}(\mathbf{Y}_s)\mathrm{d}s + \sqrt{\mathrm{K}}\mathrm{d}\mathbf{B}_s \tag{52}$$

admits a solution $(\mathbf{Y}_s)_{s\geq 0}$ whose law $\mathcal{L}(\mathbf{Y}_s)$ converges with geometric rate to p_{T-t} for any invertible matrix K.

B.6 Likelihood evaluation

Similarly to Song et al. (2021), we can derive a deterministic process which has the same marginal density as the SDE (3), which is given by the following Ordinary Differential Equation (ODE)—referred as the probability flow ODE

$$\mathrm{d}\begin{pmatrix}\mathbf{Y}_t(x)\\ \log p_t(\mathbf{Y}_t(x))\end{pmatrix} = \begin{pmatrix}\frac{1}{2}\left\{m(x) - \mathbf{Y}_t(x) - \mathrm{K}(x,x)\nabla \log p_t(\mathbf{Y}_t(x))\right\}\beta_t\\ -\frac{1}{2}\mathrm{div}\left\{m(x) - \mathbf{Y}_t(x) - \mathrm{K}(x,x)\nabla \log p_t(\mathbf{Y}_t(x))\right\}\beta_t\end{pmatrix}\mathrm{d}t. \tag{53}$$

Once the score network is learnt, we can thus use it in conjunction with an ODE solver to compute the likelihood of the model.

B.7 Discussion consistency

So far we have defined a generative model over functions via its finite marginals $\bar{\mathbf{Y}}_T^{\theta}(x)$. These finite marginals were to arise from a stochastic process if, as per the Kolmogorov extension theorem (Øksendal, 2003), they satisfy *exchangeability* and *consistency* conditions. Exchangeability can be satisfied by parametrising the score network such that the score network is equivariant w.r.t permutation, i.e. $\mathbf{s}_\theta(t, \sigma \circ x, \sigma \circ y) = \sigma \circ \mathbf{s}_\theta(t, x, y)$ for any $\sigma \in \Sigma_n$. Additionally, we have that the noising process $(\mathbf{Y}_t(x))_{x\in\mathcal{X}}$ is trivially consistent for any $t \in \mathbb{R}_+$ since it is a stochastic process as per Prop. B.1, and consequently so is the (true) time-reversal $(\bar{\mathbf{Y}}_t(x))_{x\in\mathcal{X}}$. Yet, when approximating the score $\mathbf{s}_\theta \approx \nabla \log p_t$, we lose the consistency over the generative process $\bar{\mathbf{Y}}_t^{\theta}(x)$ as the constraint on the score network is non trivial to satisfy. This is actually a really strong constraint on the model class, and as soon as one goes beyond linearity (of the posterior w.r.t. the context set), it is non trivial to enforce without directly parameterising a stochastic process, e.g. as Phillips et al. (2022). There thus seems to be a strong trade-off between satisfying consistency, and the model's ability to fit complex process and scale to large datasets.

C Manifold-valued diffusion process

C.1 Manifold-valued inputs

In the main text we dealt with a simplified case of tensor fields where the tensor fields are over Euclidean space. Nevertheless, it is certainly possible to apply our methods to these settings. Significant work has been done on performing convolutions on feature fields on generic manifolds (a superset of tensor fields on generic manifolds), core references being (Cohen, 2021) for the case of homogeneous spaces and (Weiler et al., 2021) for more general Riemannian manifolds. We

recommend these as excellent mathematical introductions to the topic and build on them to describe how to formulate diffusion models over these spaces.

Tensor fields as sections of bundles. Formally the fields we are interested in modelling are sections σ of associated tensor bundles of the principle G-bundle on a manifold M. We shall denote such a bundle BM and the space of sections $\Gamma(BM)$. The goal, therefore, is to model *random elements* from this space of sections. For a clear understanding of this definition, please see Weiler et al. (2021, pages 73-95) for an introduction suitable to ML audiences. Prior work looking at this setting is (Hutchinson et al., 2021) where they construct Gaussian Processes over tensor fields on manifolds.

Stochastic processes on spaces of sections. Given we can see sections as maps $\sigma : M \to BM$, where an element in BM is a tuple (m, b), m in the base manifold and b in the typical fibre, alongside the condition that the composition of the projection $\text{proj}_i : (m, b) \mapsto m$ with the section is the identity, $\text{proj}_i \circ \sigma = \text{Id}$ it is clear we can see distribution over sections as stochastic processes with index set the manifold M, and output space a point in the bundle BM, with the projection condition satisfied. The projection onto finite marginals, i.e. a finite set of points in the manifold, is defined as $\pi_{m_1,\dots,m_n}(\sigma) = (\sigma(m_1), ..., \sigma(m_n))$.

Noising process. To define a noising process over these marginals, we can use Gaussian Processes defined in (Hutchinson et al., 2021) over the tensor fields. The convergence results of Phillips et al. (2022) hold still, and so using these Gaussian Processes as noising processes on the marginals also defines a noising process on the whole section.

Reverse process. The results of (Cattiaux et al., 2021) are extremely general and continue to hold in this case of SDEs on the space of sections. Note we don't actually need this to be the case, we can just work with the reverse process on the marginals themselves, which are much simpler objects. It is good to know that it is a valid process on full sections though should one want to try and parameterise a score function on the whole section akin to some other infinite-dimension diffusion models.

Score function. The last thing to do therefore is parameterise the score function on the marginals. If we were trying to parameterise the score function over the *whole* section at once (akin to a number of other works on infinite dimension diffusions), this could present some problems in enforcing the smoothness of the score function. As we only deal with the score function on a finite set of marginals, however, we need not deal with this issue and this presents a distinct advantage in simplicity for our approach. All we need to do is pick a way of numerically representing points on the manifold and b) pick a basis for the tangent space of each point on the manifold. This lets us represent elements from the tangent space numerically, and therefore also elements from tensor space at each point numerically as well. This done, we can feed these to a neural network to learn to output a numerical representation of the score on the same basis at each point.

C.2 Manifold-valued outputs

In the setting, where one aim to model a stochastic process with manifold codomain $\mathbf{Y}_t(x) = (\mathbf{Y}_t(x_1), \cdots, \mathbf{Y}_t(x_n)) \in \mathcal{M}^n$, things are less trivial as manifolds do not have a vector space structure which is necessary to define Gaussian processes. Fortunately, We can still target a know distribution marginally independently on each marginal, since this is well defined, and as such revert to the Riemannian diffusion models introduced in De Bortoli et al. (2021) with n independent Langevin noising processes

$$\mathrm{d}\mathbf{Y}_t(x_k) = -\tfrac{1}{2}\nabla U(\mathbf{Y}_t(x_k))\, \beta_t \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{B}_t^{\mathcal{M}} \,. \tag{54}$$

are applied to each marginal. Hence in the limit $t \to \infty$, $\mathbf{Y}_t(x)$ has density (assuming it exists) which factors as $dp/d\text{Vol}_{\mathcal{M}}((y(x_1), \cdots, y(x_n))) \propto \prod_k e^{-U(y(x_n)))}$. For compact manifolds, we can target the uniform distribution by setting $U(x) = 0$. The reverse time process will have correlation between different marginals, and so the score function still needs to be a function of all the points in the marginal of interest.

D Invariant neural diffusion processes

D.1 $\mathrm{E}(n)$-equivariant kernels

A kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is equivariant if it satisfies the following constraints: (a) k is *stationnary*, that is if for all $x, x' \in \mathbb{R}^n$

$$k(x, x') = k(x - x') \triangleq \tilde{k}(x - x') \tag{55}$$

and if (b) it satisfies the *angular constraint* for any $h \in H$

$$k(hx, hx') = \rho(h)k(x, x')\rho(h)^\top. \tag{56}$$

A trivial example of such an equivariant kernel is the diagonal kernel $k(x, x') = k_0(x, x')\mathrm{I}$ (Holderrieth et al., 2021), with k_0 stationnary. This kernel can be understood has having d independent Gaussian process uni-dimensional output, that is, there is no inter-dimensional correlation.

Less trivial examples, are the $\mathrm{E}(n)$ equivariant kernels proposed in Macêdo and Castro (2010). Namely curl-free and divergence-free kernels, allowing for instance to model electric or magnetic fields. Formally we have $k_{\mathrm{curl}} = k_0 A$ and $k_{\mathrm{div}} = k_0 B$ with k_0 stationary, e.g. squared exponential kernel $k_0(x, x') = \sigma^2 \exp\left(\frac{\|x-x'\|^2}{2l^2}\right)$, and A and B given by

$$A(x, x') = \mathrm{I} - \frac{(x - x')(x - x')^\top}{l^2} \tag{57}$$

$$B(x, x') = \frac{(x - x')(x - x')^\top}{l^2} + \left(n - 1 - \frac{\|x - x'\|^2}{l^2}\right)\mathrm{I}. \tag{58}$$

See Holderrieth et al. (Appendix C, 2021) for a proof.

D.2 Proof of Prop. 3.2

Below we give two proofs for the group invariance of the generative process, one via the probability flow ODE and one directly via Fokker-Planck.

Proof. Reverse ODE. The reverse probability flow associated with the forward SDE (3) with approximate score $\mathbf{s}_\theta(t, \cdot) \approx \nabla \log p_t$ is given by

$$\mathrm{d}\bar{\mathbf{Y}}_t | x = \tfrac{1}{2}\left[-m(x) + \bar{\mathbf{Y}}_t + \mathrm{K}(x, x)\mathbf{s}_\theta(T - t, x, \bar{\mathbf{Y}}_t)\right]\mathrm{d}t \tag{59}$$

$$\triangleq b_{\mathrm{ODE}}(t, x, \bar{\mathbf{Y}}_t)\mathrm{d}t \tag{60}$$

This ODE induces a flow $\phi_t^b : X^n \times Y^n \to \mathrm{T}Y^n$ for a given integration time t, which is said to be G-equivariant if the vector field is $G-$equivariant itself, i.e. $b(t, g \cdot x, \rho(g)\bar{\mathbf{Y}}_t) = \rho(g)b(t, x, \bar{\mathbf{Y}}_t)$. We have that for any $g \in G$

$$b_{\mathrm{ODE}}(t, g \cdot x, \rho(g)\bar{\mathbf{Y}}_t) = \tfrac{1}{2}\left[-m(g \cdot x) + \rho(g)\mathbf{Y}_t + \mathrm{K}(g \cdot x, g \cdot x)\ \mathbf{s}_\theta(t, g \cdot x, \rho(g)\bar{\mathbf{Y}}_t)\right] \tag{61}$$

$$\overset{(1)}{=} \tfrac{1}{2}\left[-\rho(g)m(x) + \rho(g)\mathbf{Y}_t + \rho(g)\mathrm{K}(x, x)\rho(g)^\top\ \mathbf{s}_\theta(t, g \cdot x, \rho(g)\bar{\mathbf{Y}}_t)\right] \tag{62}$$

$$\overset{(2)}{=} \tfrac{1}{2}\left[-\rho(g)m(x) + \rho(g)\mathbf{Y}_t + \rho(g)\mathrm{K}(x, x)\rho(g)^\top\rho(g)\ \mathbf{s}_\theta(t, x, \bar{\mathbf{Y}}_t)\right] \tag{63}$$

$$\overset{(3)}{=} \tfrac{1}{2}\rho(g)\left[-m(x) + \mathbf{Y}_t + \mathrm{K}(x, x)\ \mathbf{s}_\theta(t, x, \bar{\mathbf{Y}}_t)\right] \tag{64}$$

$$= \rho(g)b_{\mathrm{ODE}}(t, x, \bar{\mathbf{Y}}_t) \tag{65}$$

with (1) from the G-invariant prior GP conditions on m and k, (2) assuming that the score network is G-equivariant and (3) assuming that $\rho(g) \in O(n)$. To prove the opposite direction, we can simply follow these computations backwards. Finally, we know that with a G-invariant probability measure p_{ref} and G-equivariant map ϕ, the pushforward probability measure $p_{\mathrm{ref}}^{-1} \circ \phi$ is also G-invariant (Köhler et al., 2020; Papamakarios et al., 2019). Assuming a G-invariant prior GP, and a G-equivariant score network, we thus have that the generative model from Sec. 3 defines marginals that are G-invariant. □

Proof. Reverse SDE. The reverse SDE associated of the forward SDE (3) with approximate score $\mathbf{s}_\theta(t, \cdot) \approx \nabla \log p_t$ is given by

$$\mathrm{d}\bar{\mathbf{Y}}_t | x = \left[-(m(x) - \bar{\mathbf{Y}}_t)/2 + \mathrm{K}(x, x)\mathbf{s}_\theta(T - t, x, \bar{\mathbf{Y}}_t)\right]\mathrm{d}t + \sqrt{\beta_t \mathrm{K}(x, x)}\mathrm{d}\mathbf{B}_t \tag{66}$$

$$\triangleq b_{\mathrm{SDE}}(t, x, \bar{\mathbf{Y}}_t)\mathrm{d}t + \Sigma^{1/2}(t, x)\ \mathrm{d}\mathbf{B}_t. \tag{67}$$

As for the probability flow drift b_{ODE}, we have that b_{SDE} is similarly G-equivariant, that is $b_{\text{SDE}}(t, g \cdot x, \rho(g)\bar{\mathbf{Y}}_t) = \rho(g) b_{\text{SDE}}(t, x, \bar{\mathbf{Y}}_t)$ for any $g \in G$. Additionally, we have that diffusion matrix is also G-equivariant as for any $g \in G$ we have $\Sigma(t, g \cdot x) = \beta_t \mathrm{K}(g \cdot x, g \cdot x) = \beta_t \rho(g) \mathrm{K}(x, x) \rho(g)^\top = \rho(g) \Sigma(t, x) \rho(g)^\top$ since K is the gram matrix of an G-equivariant kernel k.

Additionally assuming that b_{SDE} and Σ are bounded, Yim et al. (Proposition 3.6, 2023) says that the distribution of $\bar{\mathbf{Y}}_t$ is G-invariant, and in in particular $\mathcal{L}(\bar{\mathbf{Y}}_0)$.

□

D.3 Equivariant posterior maps

Theorem D.1 (Invariant prior stochastic process implies an equivariant posterior map)**.** *Using the language of Weiler et al. (2021) our tensor fields are sections of an associated vector bundle $\mathcal{A}$ of a manifold M with a G structure. Let Isom_{GM} be the group of G-structure preserving isometries on M. The action of this group on a section of the bundle $f \in \Gamma(\mathcal{A})$ is given by*

$$\phi \triangleright f := \phi_{*,\mathcal{A}} \circ f \circ \phi^{-1}$$

(Weiler et al., 2021). Let $f \sim P$, P a distribution over the space of section. Let $\phi \triangleright P$ be the law of of $\phi \triangleright f$. Let $\mu_x = \mathcal{L}(f(x)) = \pi_{x\#} P$, the law of f evaluated at a point, where π_x is the canonical projection operator onto the marginal at x, $\#$ the pushforward operator in the measure theory sense, $x \in M$ and y is in the fibre of the associated bundle. Let $\mu_x^{x',y} = \mathcal{L}(f(x) | f(x') = y') = \pi_x \mu^{x',y'} = \pi_{x\#} \mathcal{L}(f | f(x') = y')$, the conditional law of the process when given $f(x') = y'$.

Assume that the prior is invariant under the action of Isom_{GM}, i.e. that

$$\phi \triangleright \mu_x = (\phi_{*,\mathcal{A}})_{\#} \mu_{\phi^{-1}(x)} = \mu_x$$

Then the conditional measures are equivariant, in the sense that

$$\phi \triangleright \mu_x^{x',y'} = (\phi_{*,\mathcal{A}})_{\#} \mu_{\phi^{-1}(x)}^{x',y'} = \mu_x^{\phi^{-1}(x), \phi_{*,\mathcal{A}}(y)} = \mu_x^{\phi \triangleright (x',y')}$$

Proof. $\forall A, B$ test functions, $\phi \in \mathrm{Isom}_{GM}$,

$$\begin{aligned}
\mathrm{E}[B(f(x'))A((\phi \triangleright f)(x))] &= \mathrm{E}\big[B(f(x'))A\big(\phi_{*,\mathcal{A}} \circ f \circ \phi^{-1}(x)\big)\big] \\
&= \mathrm{E}\big[B(f(x'))\mathrm{E}\big[A\big(\phi_{*,\mathcal{A}}\big(F(\phi^{-1}(x))\big)\big) \;\big|\; F(x')\big]\big] \\
&= \mathrm{E}\left[B(f(x')) \int A(y) (\phi_{*,\mathcal{A}})_{\#} \mu_{\phi^{-1}(x)}^{x', f(x')}(\mathrm{d}y)\right] \\
&= \int B(y') \int A(y) (\phi_{*,\mathcal{A}})_{\#} \mu_{\phi^{-1}(x)}^{x', f(x')}(\mathrm{d}y) \mu_{x'}(\mathrm{d}y') \\
&= \int B(y') \int A(y) \Big(\phi \triangleright \mu_x^{x', f(x')}\Big)(\mathrm{d}y) \mu_{x'}(\mathrm{d}y')
\end{aligned}$$

By invariance this quantity is also equal to

$$\begin{aligned}
\mathrm{E}\big[B\big((\phi^{-1} \triangleright f)(x')\big)A((\phi^{-1} \triangleright \phi \triangleright f)(x))\big] &= \mathrm{E}\big[B\big((\phi^{-1} \triangleright f)(x')\big)\mathrm{E}\big[A(f(x)) \;\big|\; B\big((\phi^{-1} \triangleright f)(x')\big)\big]\big] \\
&= \mathrm{E}\big[B\big(\phi_{*,\mathcal{A}}(f(\phi^{-1}(x')))\big)\big[A(F(x)) \;\big|\; \phi_{*,\mathcal{A}}(f(\phi^{-1}(x')))\big]\big] \\
&= \mathrm{E}\left[B\Big(\tau_{x',g}^{-1} F(gx')\Big) \int A(y) \mu_x^{\phi(x'), \phi_{*,\mathcal{A}}^{-1}(y)}\right](\mathrm{d}y) \\
&= \int B(y') \int A(y) \mu_x^{\phi \triangleright (x',y)}(\mathrm{d}y) \Big(\phi_{*,\mathcal{A}}^{-1}\Big)_{\#} \mu_{\phi(x')}(\mathrm{d}y') \\
&= \int B(y') \int A(y) \mu_x^{\phi \triangleright (x',y)}(\mathrm{d}y) \big(\phi^{-1} \triangleright \mu_{x'}\big)(\mathrm{d}y')
\end{aligned}$$

Hence

$$\Big(\phi \triangleright \mu_x^{x', f(x')}\Big)(\mathrm{d}y) \mu_{x'}(\mathrm{d}y') = \mu_x^{\phi \triangleright (x',y)}(\mathrm{d}y) \big(\phi^{-1} \triangleright \mu_{x'}\big)(\mathrm{d}y')$$

By the stated invariance $\phi^{-1} \triangleright \mu_{x'} = \mu_{x'}$, hence

$$\left(\phi \triangleright \mu_x^{x', f(x')}\right)(\mathrm{d}y) = \mu_x^{\phi \triangleright (x', y)}(\mathrm{d}y) \text{ a.e. } y'$$

So

$$\phi \triangleright \mu_x^{x', f(x')} = \mu_x^{\phi \triangleright (x', y)} \tag{68}$$

as desired. □

E Langevin corrector and the iterative procedure of REPAINT (Lugmayr et al., 2022)

E.1 Langevin sampling scheme

Several previous schemes exist for conditional sampling from Diffusion models. Two different types of conditional sampling exist. Those that try to sample conditional on some part of the state space over which the diffusion model has been trained, such as in-painting or extrapolation tasks, and those that post-hoc attempt to condition on something outside the state space that the model has been trained on.

This first category is the one we are interested in, and in it we have:

- Replacement sampling (Song et al., 2021), where the reverse ODE or SDE is evolved but by fixing the conditioning data during the rollout. This method does produce visually coherent sampling in some cases, but is not an exact conditional sampling method.
- SMC-based methods (Trippe et al., 2022), which are an exact method up to the particle filter assumption. These can produce good results but can suffer from the usual SMC methods downsides on highly multi-model data such as particle diversity collapse.
- The RePaint scheme of (Lugmayr et al., 2022). While not originally proposed as an exact sampling scheme, we will show later that it can in fact be shown that this method is doing a specific instantiation of our newly proposed method, and is therefore exact.
- Amortisation methods, e.g. Phillips et al. (2022). While they can be effective, these methods can never perform exact conditional sampling, by definition.

Our goal is to produce an exact sampling scheme that does not rely on SMC-based methods. Instead, we base our method on Langevin dynamics. If we have a score function trained over the state space $\boldsymbol{x} = [\boldsymbol{x}^c, \boldsymbol{x}^*]$, where $\boldsymbol{x}^c$ are the points we wish to condition on and $\boldsymbol{x}_s$ points we wish to sample, we exploit the following score breakdown:

$$\nabla_{\boldsymbol{x}^*} \log p(\boldsymbol{x}^* | \boldsymbol{x}^c) = \nabla_{\boldsymbol{x}^*} \log p([\boldsymbol{x}^*, \boldsymbol{x}^c]) - \nabla_{\boldsymbol{x}^*} \log p(\boldsymbol{x}^c) = \nabla_{\boldsymbol{x}^*} \log p(\boldsymbol{x})$$

If we have access to the score on the joint variables, we, therefore, have access to the conditional score by simply only taking the gradient of the joint score for the variable we are not conditioning on.

Given we have learnt $s_\theta(t, \boldsymbol{x}) \approx \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, we could use this to perform Langevin dynamics at $t = \epsilon$, some time very close to 0. Similar to (Song and Ermon, 2019) however, this produces the twin issues of how to initialise the dynamics, given a random initialisation will start the sampler in a place where the score has been badly learnt, producing slow and inaccurate sampling.

Instead, we follow a scheme of tempered Langevin sampling detailed in Alg. 1. Starting at $t = T$ we sample an initialisation of $\boldsymbol{y}^*$ based on the reference distribution. Progressing from $t = T$ towards $t = \epsilon$ we alternate between running a series of Lavgevin corrector steps to sample from the distribution $p_{t,\boldsymbol{x}^*}(\boldsymbol{y}^* | \boldsymbol{y}^c)$, and a single backwards SDE step to sample from $p_{\boldsymbol{x}}(\boldsymbol{y}_{t-\gamma} | \boldsymbol{y}_t)$ with a step size γ. At each inner and outer step, we sample a noised version of the conditioning points $\boldsymbol{y}^c$ based forward SDE applying noise to these context points, $p_{t,\boldsymbol{x}^c}(\boldsymbol{y}_t^c | \boldsymbol{y}^c)$. For the exactness of this scheme, all that matters is that at the end of the sampling scheme, we are sampling from $p_{\boldsymbol{x}^*}(\boldsymbol{y}^* | \boldsymbol{y}^c)$ (up to the ϵ away from zero clipping of the SDE). The rest of the scheme is designed to map from the initial sample at $t = T$ of $\boldsymbol{y}^*$ to a viable sample through *regions where the score has been learnt well*.

Given the noising scheme applied to the context points does not actually play into the theoretical exactness of the scheme, only the practical difficulty of staying near regions of well-learnt score, we could make a series of different choices for how to noise the context set at each step.

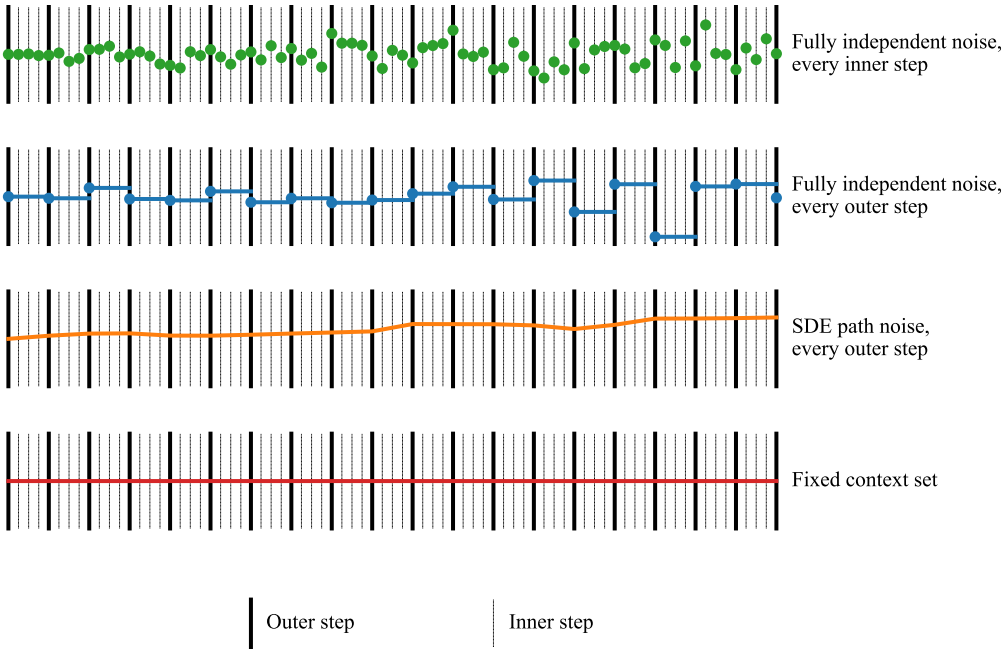


Figure 8: Comparison of different context noising schemes for the conditional sampling.

Table 4: Comparison of complexity of different noise sampling schemes for the context set.

Scheme	Closed-form noise	Simulated noise
Re-sample noise at every inner step	$\mathcal{O}(NI)$	$\mathcal{O}(N^2I^2)$
Re-sample noise at every outer step	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$
Sampling an SDE path on the context	$\mathcal{O}(N)$	$\mathcal{O}(N)$
No noise applied	-	-

The choices that present themselves are

1. The initial scheme of sampling context noise from the SDE every inner and outer step.
2. Only re-sampling the context noise every outer step, and keeping it fixed to this for each inner step associated with the outer step.
3. Instead of sampling independent marginal noise at each outer step, sampling a single noising trajectory of the context set from the forward SDE and use this as the noise at each time.
4. Perform no noising at all. Effectively the replacement method with added Langevin sampling.

These are illustrated in Fig. 8. The main trade-off of different schemes is the speed at which the noise can be sampled vs sample diversity. In the Euclidean case, we have a closed form for the evolution of the marginal density of the context point under the forward SDE. In this case sampling the noise at a given time is $\mathcal{O}(1)$ cost. On the other hand, in some instances such as nosing SDEs on general manifolds, we have to simulate this noise by discretising the forward SDE. In this case, it is $\mathcal{O}(n)$ cost, where n is the number of discretisation steps in the SDE. For N outer steps and I inner steps, the complexity of the different noising schemes is compared in Table 4. Note the conditional sampling scheme other than the noise sampling is $\mathcal{O}(NI)$ complexity.

E.2 RePaint (Lugmayr et al., 2022) correspondance

In this section, we show that:

Algorithm 1 Conditional sampling with Langevin dynamics.

Require: Score network $\mathbf{s}_\theta(t, \boldsymbol{x}, \boldsymbol{y})$, conditioning points $(\boldsymbol{x}^c, \boldsymbol{y}^c)$, query locations $\boldsymbol{x}^*$
$\bar{\boldsymbol{x}} = [\boldsymbol{x}^c, \boldsymbol{x}^*]$ ▷ Augmented inputs set
$\tilde{\boldsymbol{y}}^*_T \sim \mathrm{N}(m(\boldsymbol{x}^*), k(\boldsymbol{x}^*, \boldsymbol{x}^*))$ ▷ Sample initial noise
for $t \in \{T, T-\gamma, ..., \epsilon\}$ **do**
  $\boldsymbol{y}^c_t \sim p_{t,\boldsymbol{x}^c}(\boldsymbol{y}^c_t | \boldsymbol{y}^c_0)$ ▷ Noise context outputs
  $Z \sim \mathcal{N}(0, \mathrm{Id})$ ▷ Sample tangent noise
  $\left[\_, \tilde{\boldsymbol{y}}^*_{t-\gamma}\right] = [\boldsymbol{y}^c_t, \boldsymbol{y}^*_t] + \gamma \left\{-\frac{1}{2}\left(m(\bar{\boldsymbol{x}}) - [\boldsymbol{y}^c_t, \boldsymbol{y}^*_t]\right) + \mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})\mathbf{s}_\theta(t, \bar{\boldsymbol{x}}, [\boldsymbol{y}^c_t, \tilde{\boldsymbol{y}}^*_t])\right\} + \sqrt{\gamma}\mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{1/2} Z$ ▷ Euler-Maruyama step
  for $l \in \{1, \dots, L\}$ **do**
    $\boldsymbol{y}^c_{t-\gamma} \sim p_{t-\gamma,\boldsymbol{x}^c}(\boldsymbol{y}^c_{t-\gamma} | \boldsymbol{y}^c_0)$ ▷ Noise context outputs
    $Z \sim \mathcal{N}(0, \mathrm{Id})$ ▷ Sample tangent noise
    $\left[\_, \tilde{\boldsymbol{y}}^*_{t-\gamma}\right] = \left[\_, \tilde{\boldsymbol{y}}^*_{t-\gamma}\right] + \frac{\gamma}{2}\mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})\mathbf{s}_\theta(t-\gamma, \bar{\boldsymbol{x}}, \left[\boldsymbol{y}^c_{t-\gamma}, \tilde{\boldsymbol{y}}^*_{t-\gamma}\right]) + \sqrt{\gamma}\mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{1/2} Z$ ▷ Langevin step
  $\boldsymbol{y}^*_{t-\gamma} = \tilde{\boldsymbol{y}}^*_{t-\gamma}$
return $\boldsymbol{y}^*_\epsilon$

(a) Alg. 1 and Alg. 2 `Repaint` from (Lugmayr et al., 2022) are equivalent in a specific setting.

(b) There exists a continuous limit (SDE) for both procedures. This SDE targets a probability density which *does not* correspond to $p(x_{t_0}|x^c_0)$.

(c) When $t_0 \to 0$ this probability measure converges to $p(x_0|x^c_0)$ which ensures the correctness of the proposed sampling scheme.

We begin by recalling the conditional sampling algorithm we study in Alg. 1 and Alg. 2.

Algorithm 2 REPAINT (Lugmayr et al., 2022).

Require: Score network $\mathbf{s}_\theta(t, \boldsymbol{x}, \boldsymbol{y})$, conditioning points $(\boldsymbol{x}^c, \boldsymbol{y}^c)$, query locations $\boldsymbol{x}^*$
$\bar{\boldsymbol{x}} = [\boldsymbol{x}^c, \boldsymbol{x}^*]$ ▷ Augmented inputs set
$[\boldsymbol{y}^c_T, \boldsymbol{y}^*_T] \sim \mathrm{N}(m(\bar{\boldsymbol{x}}), k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}))$ ▷ Sample initial noise
for $t \in \{T, T-\gamma, ..., \epsilon\}$ **do**
  $\tilde{\boldsymbol{y}}^*_t = \boldsymbol{y}^*_t$
  for $l \in \{1, \dots, L\}$ **do**
    $\boldsymbol{y}^c_t \sim \mathrm{N}(m_t(\boldsymbol{x}^c; \boldsymbol{y}^c), k_t(\boldsymbol{x}^c, \boldsymbol{x}^c; \boldsymbol{y}^c))$ ▷ Noise context outputs
    $Z \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Sample tangent noise
    $\left[\_, \tilde{\boldsymbol{y}}^*_{t-\gamma}\right] = [\boldsymbol{y}^c_t, \tilde{\boldsymbol{y}}^*_t] + \gamma \left\{-\frac{1}{2}\left(m(\bar{\boldsymbol{x}}) - [\boldsymbol{y}^c_t, \tilde{\boldsymbol{y}}^*_t]\right) + \mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})\mathbf{s}_\theta(t, \bar{\boldsymbol{x}}, [\boldsymbol{y}^c_t, \tilde{\boldsymbol{y}}^*_t])\right\} + \sqrt{\gamma}\mathrm{K}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})^{1/2} Z$ ▷ Reverse step
    $Z \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Sample tangent noise
    $\tilde{\boldsymbol{y}}^*_t = \tilde{\boldsymbol{y}}^*_{t-\gamma} + \gamma \left\{\frac{1}{2}\left(m(\boldsymbol{x}^*) - \tilde{\boldsymbol{y}}^*_{t-\gamma}\right)\right\} + \sqrt{\gamma}\mathrm{K}(\boldsymbol{x}^*, \boldsymbol{x}^*)^{1/2} Z$ ▷ Forward step
  $\boldsymbol{y}^*_{t-\gamma} = \tilde{\boldsymbol{y}}^*_{t-\gamma}$
return $\boldsymbol{y}^*_\epsilon$

First, we start by describing the RePaint algorithm (Lugmayr et al., 2022). We consider $(Z^0_k, Z^1_k, Z^2_k)_{k\in\mathbb{N}}$ a sequence of independent Gaussian random variable such that for any $k \in \mathbb{N}$, Z^1_k and Z^2_k are d-dimensional Gaussian random variables with zero mean and identity covariance matrix and Z^0_k is a p-dimensional Gaussian random variable with zero mean and identity covariance matrix. We assume that the whole sequence to be inferred is of size d while the context is of size p. For simplicity, we only consider the Euclidean setting with $\mathrm{K} = \mathrm{Id}$. The proofs can be adapted to cover the case $\mathrm{K} \neq \mathrm{Id}$ without loss of generality.

Let us fix a time $t_0 \in [0, T]$. We consider the chain $(X_k)_{k\in\mathbb{N}}$ given by $X_0 \in \mathbb{R}^d$ and for any $k \in \mathbb{N}$, we define

$$X_{k+1/2} = \mathrm{e}^\gamma X_k + 2(\mathrm{e}^\gamma - 1)\nabla_{x_k} \log p_{t_0}([X_k, X^c_k]) + (\mathrm{e}^{2\gamma} - 1)^{1/2} Z^1_k, \tag{69}$$

where $X^c_k = \mathrm{e}^{-t_0} X^c_0 + (1 - \mathrm{e}^{-2t_0})^{1/2} Z^0_k$. Finally, we consider

$$X_{k+1} = \mathrm{e}^{-\gamma} X_{k+1/2} + (1 - \mathrm{e}^{-2\gamma})^{1/2} Z^2_k. \tag{70}$$

Note that (69) corresponds to one step of *backward SDE* integration and (70) corresponds to one step of *forward SDE* integration. In both cases we have used the exponential integrator, see (De

Bortoli, 2022) for instance. While we use the exponential integrator in the proofs for simplicity other integrators such as the classical Euler-Maruyama integration could have been used. Combining (69) and (70), we get that for any $k \in \mathbb{N}$ we have

$$X_{k+1} = X_k + 2(1 - \mathrm{e}^{-\gamma})\nabla_{x_k} \log p_{t_0}([X_k, X_k^c]) + (1 - \mathrm{e}^{-2\gamma})^{1/2}(Z_k^1 + Z_k^2). \tag{71}$$

Remarking that $(Z_k)_{k\in\mathbb{N}} = ((Z_k^1 + Z_k^3)/\sqrt{2})_{k\in\mathbb{N}}$ is a family of d-dimensional Gaussian random variables with zero mean and identity covariance matrix, we get that for any $k \in \mathbb{N}$

$$X_{k+1} = X_k + 2(1 - \mathrm{e}^{-\gamma})\nabla_{x_k} \log p_{t_0}([X_k, X_k^c]) + \sqrt{2}(1 - \mathrm{e}^{-2\gamma})^{1/2} Z_k, \tag{72}$$

where we recall that $X_k^c = \mathrm{e}^{-t_0} X_0^c + (1 - \mathrm{e}^{-2t_0})^{1/2} Z_k^0$. Note that the process (72) is another version of the `Repaint` algorithm (Lugmayr et al., 2022), where we have concatenated the denoising and noising procedure. With this formulation, it is clear that `Repaint` is equivalent to Alg. 1. In what follows, we identify the limitating SDE of this process.

In what follows, we describe the limiting behavior of (72) under mild assumptions on the target distribution. In what follows, for any $x_{t_0} \in \mathbb{R}^d$, we denote

$$b(x_{t_0}) = 2 \int_{\mathbb{R}^p} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x_{t_0}^c]) p_{t_0|0}(x_{t_0}^c | x_0^c) \mathrm{d}x_{t_0}^c. \tag{73}$$

We emphasize that $b/2 \neq \nabla_{x_{t_0}} \log p(\cdot | x_0^c)$. In particular, using Tweedie's identity, we have that for any $x_{t_0} \in \mathbb{R}^d$

$$\nabla \log p_{t_0}(x_{t_0} | x_0^c) = \int_{\mathbb{R}^p} \nabla_{x_{t_0}} \log p([x_{t_0}, x_{t_0}^c] | x_0^c) p(x_{t_0}^c | x_{t_0}, x_0^c) \mathrm{d}x_{t_0}^c. \tag{74}$$

We introduce the following assumption.

Assumption 1. *There exist* $\mathtt{L}, C \geq 0$, $\mathtt{m} > 0$ *such that for any* $x_{t_0}^c, y_t^c \in \mathbb{R}^p$ *and* $x_{t_0}, y_t \in \mathbb{R}^d$

$$\|\nabla \log p_{t_0}([x_{t_0}, x_{t_0}^c]) - \nabla \log p_{t_0}([y_t, y_t^c])\| \leq \mathtt{L}(\|x_{t_0} - y_t\| + \|x_{t_0}^c - y_t^c\|). \tag{75}$$

Assumption 1 ensures that there exists a unique strong solution to the SDE associated with (72). Note that conditions under which $\log p_{t_0}$ is Lipschitz are studied in De Bortoli (2022). In the theoretical literature on diffusion models the Lipschitzness assumption is classical, see Lee et al. (2023) and Chen et al. (2022).

We denote $((\mathbf{X}_t^\gamma)_{t\geq 0})_{\gamma>0}$ the family of processes such that for any $k \in \mathbb{N}$ and $\gamma > 0$, we have for any $t \in [k\gamma, (k+1)\gamma)$, $\mathbf{X}_t^\gamma = (1 - (t - k\gamma)/\gamma)\mathbf{X}_{k\gamma}^\gamma + (t - k\gamma)/\gamma \mathbf{X}_{(k+1)\gamma}^\gamma$ and

$$\mathbf{X}_{(k+1)\gamma}^\gamma = \mathbf{X}_{k\gamma}^\gamma + 2(1 - \mathrm{e}^{-\gamma})\nabla_{\mathbf{X}_{k\gamma}^\gamma} \log p_{t_0}([\mathbf{X}_{k\gamma}^\gamma, \mathbf{X}_{k\gamma}^{c,n}]) + \sqrt{2}(1 - \mathrm{e}^{-2\gamma})^{1/2} \mathbf{Z}_{k\gamma}^\gamma, \tag{76}$$

where $(\mathbf{Z}_{k\gamma}^\gamma)_{k\in\mathbb{N},\gamma>0}$ is a family of independent d-dimensional Gaussian random variables with zero mean and identity covariance matrix and for any $k \in \mathbb{N}$, $\gamma > 0$, $\mathbf{X}_{k\gamma}^{c,\gamma} = \mathrm{e}^{-t_0} x_0^c + (1 - \mathrm{e}^{-2t_0})^{1/2} \mathbf{Z}_{k\gamma}^{0,\gamma}$, where $(\mathbf{Z}_{k\gamma}^{0,\gamma})_{k\in\mathbb{N},\gamma>0}$ is a family of independent p-dimensional Gaussian random variables with zero mean and identity covariance matrix. This is a *linear interpolation* of the `Repaint` algorithm in the form of (72).

Finally, we denote $(\mathbf{X}_t)_{t\geq 0}$ such that

$$\mathrm{d}\mathbf{X}_t = b(\mathbf{X}_t)\mathrm{d}t + 2\mathbf{B}_t, \qquad \mathbf{X}_0 = x_0. \tag{77}$$

We recall that b depends on t_0 but t_0 is *fixed* here. This means that we are at time t_0 in the diffusion and consider a *corrector* at this stage. The variable t does not corresponds to the backward evolution but to the forward evolution *in the corrector stage*. Under Assumption 1, (77) admits a unique strong solution. The rest of the section is dedicated to the proof of the following result.

Theorem E.1. *Assume* Assumption 1. *Then* $\lim_{n\to+\infty} (\mathbf{X}_t^{1/n})_{t\geq 0} = (\mathbf{X}_t)_{t\geq 0}$.

This result is an application of Stroock and Varadhan (2007, Theorem 11.2.3). It explicits what is the *continuous* limit of the `Repaint` algorithm (Lugmayr et al., 2022).

In what follows, we verify that the assumptions of this result hold in our setting. For any $\gamma > 0$ and $x \in \mathbb{R}^d$, we define

$$b_\gamma(x) = (2/\gamma)[(1-\mathrm{e}^{-\gamma}) \int_{\mathbb{R}^d} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}]) p_{t_0|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0} \tag{78}$$

$$- (1/\gamma)\mathbb{E}[(\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}) \mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq 1} \, | \mathbf{X}_{k\gamma} = x], \tag{79}$$

$$\Sigma_\gamma(x) = (4/\gamma)(1-\mathrm{e}^{-\gamma})^2 \int_{\mathbb{R}^d} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}])^{\otimes 2} p_{t_0|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0} + (2/\gamma)(1-\mathrm{e}^{-2\gamma}) \, \mathrm{Id} \tag{80}$$

$$- (1/\gamma)\mathbb{E}[(\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma})^{\otimes 2} \mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq 1} \, | \mathbf{X}_{k\gamma} = x]. \tag{81}$$

Note that for any $\gamma > 0$ and $x \in \mathbb{R}^d$, we have

$$b_\gamma(x) = \mathbb{E}[\mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \leq 1} (\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}) \, | \mathbf{X}^\gamma_{k\gamma} = x] \tag{82}$$

$$\Sigma_\gamma(x) = \mathbb{E}[\mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \leq 1} (\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma})^{\otimes 2} \, | \mathbf{X}^\gamma_{k\gamma} = x] \tag{83}$$

$$\tag{84}$$

Lemma E.2. *Assume* Assumption 1. *Then, we have that for any $R, \varepsilon > 0$ and $\gamma \in (0,1)$*

$$\lim_{\gamma \to 0} \sup\{\|\Sigma_\gamma(x) - \Sigma(x)\| \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0, \tag{85}$$

$$\lim_{\gamma \to 0} \sup\{\|b_\gamma(x) - b(x)\| \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0, \tag{86}$$

$$\lim_{\gamma \to 0} (1/\gamma) \sup\{\mathbb{P}(\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq \varepsilon \, | \mathbf{X}_{k\gamma} = x) \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0. \tag{87}$$

Where we recall that for any $x \in \mathbb{R}^d$,

$$b(x) = 2 \int_{\mathbb{R}^p} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}]) p^x_{t|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0}, \qquad \Sigma(x) = 4 \, \mathrm{Id} \, . \tag{88}$$

Proof. Let $R, \varepsilon > 0$ and $\gamma \in (0,1)$. Using Assumption 1, there exists $C > 0$ such that for any $x_{t_0} \in \mathbb{R}^d$ with $\|x_{t_0}\| \leq R$, we have $\|\nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}])\| \leq C(1 + \|x^c_{t_0}\|)$. Since $p^c_{t_0|0}$ is Gaussian with zero mean and covariance matrix $(1-\mathrm{e}^{-2t_0}) \, \mathrm{Id}$, we get that for any $p \in \mathbb{N}$, there exists $A_k \geq 0$ such that for any $x_{t_0} \in \mathbb{R}^d$ with $\|x_{t_0}\| \leq R$

$$\int_{\mathbb{R}^d} \|\nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}])\|^p p^c_{t_0|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0} \leq A_k(1 + \|x^c_0\|^p). \tag{89}$$

Therefore, using this result and the fact that for any $s \geq 0$, $\mathrm{e}^{-s} \geq 1 - s$, we get that there exists $B_k \geq 0$ such that for any $k, p \in \mathbb{N}$ and for any $x_{t_0} \in \mathbb{R}^d$ with $\|x_{t_0}\| \leq R$

$$\mathbb{E}[\|\mathbf{X}_{(k+1)\gamma} - \mathbf{X}_{k\gamma}\|^p \, | \mathbf{X}_{k\gamma} = x] \leq B_k \gamma^{p/2}(1 + \|x^c_0\|^p). \tag{90}$$

Therefore, combining this result and the Markov inequality, we get that for any $x_{t_0} \in \mathbb{R}^d$ with $\|x_{t_0}\| \leq R$ we have

$$\lim_{\gamma \to 0} (1/\gamma) \sup\{\mathbb{P}(\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq \varepsilon \, | \mathbf{X}_{k\gamma} = x) \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0, \tag{91}$$

$$\lim_{\gamma \to 0} (1/\gamma) \|\mathbb{E}[(\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}) \mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq 1} \, | \mathbf{X}_{k\gamma} = x]\| = 0, \tag{92}$$

$$\lim_{\gamma \to 0} (1/\gamma) \|\mathbb{E}[(\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}) \mathbf{1}_{\|\mathbf{X}^\gamma_{(k+1)\gamma} - \mathbf{X}^\gamma_{k\gamma}\| \geq 1} \, | \mathbf{X}_{k\gamma} = x]\| = 0 \tag{93}$$

In addition, we have that for any $x_{t_0} \in \mathbb{R}^d$ with $R > 0$

$$|(2/\gamma)(1-\mathrm{e}^{-\gamma}) - 2| \| \int_{\mathbb{R}^d} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}]) p_{t_0|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0}\| \tag{94}$$

$$\leq A_1(1 + \|x^c_0\|)(2/\gamma)\big|\mathrm{e}^{-\gamma} - 1 + \gamma\big|. \tag{95}$$

We also have that for any $x_{t_0} \in \mathbb{R}^d$ with $R > 0$

$$(4/\gamma)|1-\mathrm{e}^{-\gamma}|^2 \| \int_{\mathbb{R}^d} \nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}])^{\otimes 2} p_{t_0|0}(x^c_{t_0}|x^c_0) \mathrm{d}x^c_{t_0}\| \tag{96}$$

$$\leq A_2(1 + \|x^c_0\|^2)(4/\gamma)\big|1-\mathrm{e}^{-\gamma}\big|^2. \tag{97}$$

Combining this result, (91), the fact that $\lim_{\gamma \to 0}(4/\gamma)|1-\mathrm{e}^{-\gamma}|^2 = 0$ and $\lim_{\gamma \to 0}(2/\gamma)|\mathrm{e}^{-\gamma} - 1 + \gamma| = 0$, we get that $\lim_{\gamma \to 0} \sup\{\|\Sigma_\gamma(x) - \Sigma(x)\| \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0$. Similarly, using (91), (94) and the fact that $\lim_{\gamma \to 0}(4/\gamma)|1-\mathrm{e}^{-\gamma}|^2 = 0$, we get that $\lim_{\gamma \to 0} \sup\{\|b_\gamma(x) - b(x)\| \, | x \in \mathbb{R}^d, \, \|x\| \leq R\} = 0$. □

We can now conclude the proof of Theorem E.1.

Proof. We have that $x \mapsto b(x)$ and $x \mapsto \Sigma(x)$ are continuous. Combining this result and Lemma E.2, we conclude the proof upon applying Stroock and Varadhan (2007, Theorem 11.2.3). □

Theorem E.1 is a non-quantitative result which states what is the limit chain for the REPAINT procedure. Note that if we do not resample, we get that

$$b^{\mathrm{cond}}(x) = 2\nabla_{x_{t_0}} \log p_{t_0}([x_{t_0}, x^c_{t_0}]), \qquad \Sigma(x) = 4\,\mathrm{Id}\,. \tag{98}$$

Recalling (88), we get that (98) is an *amortised version* of b^{cond}. Similar convergence results can be derived in this case. Note that it is also possible to obtain quantitative discretization bounds between $(\mathbf{X}_t)_{t\geq 0}$ and $(\mathbf{X}^{1/n}_t)_{t\geq 0}$ under the ℓ^2 distance. These bounds are usually leveraged using the Girsanov theorem (Durmus and Moulines, 2017; Dalalyan, 2017). We leave the study of such bounds for future work.

We also remark that $b(x_{t_0})$ is *not* given by $\nabla \log p_{t_0}(x_{t_0}|x^c_0)$. Denoting U_{t_0} such that for any $x_{t_0} \in \mathbb{R}^d$

$$U_{t_0}(x_{t_0}) = -\int_{\mathbb{R}^p} (\log p_{t_0}(x_{t_0}|x^c_{t_0}))p_{t|0}(x^c_{t_0}|x^c_0)\mathrm{d}x^c_{t_0}, \tag{99}$$

we have that $\nabla U_{t_0}(x_{t_0}) = -b(x_{t_0})$, under mild integration assumptions. In addition, using Jensen's inequality, we have

$$\int_{\mathbb{R}^d} \exp[-U_{t_0}(x_{t_0})]\mathrm{d}x_{t_0} \leq \int_{\mathbb{R}^d}\int_{\mathbb{R}^p} p_{t_0}(x_{t_0}|x^c_{t_0})p_{t|0}(x^c_{t_0}|x^c_0)\mathrm{d}x_{t_0}\mathrm{d}x^c_{t_0} \leq 1. \tag{100}$$

Hence, π_{t_0} with density proportional to $x \mapsto \exp[-U_{t_0}(x)]$ defines a valid probability measure.

We make the following assumption which allows us to control the ergodicity of the process $(\mathbf{X}_t)_{t\geq 0}$.

Assumption 2. *There exist* $\mathtt{m} > 0$ *and* $C \geq 0$ *such that for any* $x_{t_0} \in \mathbb{R}^d$ *and* $x^c_{t_0} \in \mathbb{R}^p$

$$\langle \nabla_{x_t} \log p_{t_0}([x_t, x^c_t]), x_t\rangle \leq -\mathtt{m}\|x_t\|^2 + C(1 + \|x^c_t\|^2). \tag{101}$$

The following proposition ensures the ergodicity of the chain $(\mathbf{X}_t)_{t\geq 0}$. It is a direct application of Roberts and Tweedie (1996, Theorem 2.1).

Proposition E.3. *Assume* Assumption 1 *and* Assumption 2. *Then,* π_{t_0} *is the unique invariant probability measure of* $(\mathbf{X}_t)_{t\geq 0}$ *and* $\lim_{t\to 0} \|\mathcal{L}(\mathbf{X}_t) - \pi_{t_0}\|_{\mathrm{TV}} = 0$, *where* $\mathcal{L}(\mathbf{X}_t)$ *is the distribution of* $\mathbf{X}_t$.

Finally, for any $t_0 > 0$, denoting π_{t_0} the probability measure with density U_{t_0} given for any $x_{t_0} \in \mathbb{R}^d$ by

$$U_{t_0}(x_{t_0}) = -\int_{\mathbb{R}^p} (\log p_{t_0}(x_{t_0}|x^c_{t_0}))p_{t|0}(x^c_{t_0}|x^c_0)\mathrm{d}x^c_{t_0}. \tag{102}$$

We show that the family of measures $(\pi_{t_0})_{t_0>0}$ approximates the posterior with density $x_0 \mapsto p(x_0|x^c_0)$ when t_0 is small enough.

Proposition E.4. *Assume* Assumption 1. *We have that* $\lim_{t_0\to 0} \pi_{t_0} = \pi_0$ *where* π_0 *admits a density w.r.t. the Lebesgue measure given by* $x_0 \mapsto p(x_0|x^c_0)$.

Proof. This is a direct consequence of the fact that $p_{t|0}(\cdot|x^c_0) \to \delta_{x^c_0}$. □

This last results shows that even though we do not target $x_{t_0} \mapsto p_{t_0|0}(x_{t_0}|x^c_0)$ using this corrector term, we still target $p(x_0|x^c_0)$ as $t_0 \to 0$ which corresponds to the desired output of the algorithm.

F Experimental details

Models, training and evaluation have been implemented in `Jax` (Bradbury et al., 2018). We used Python (Van Rossum and Drake Jr, 1995) for all programming, Hydra (Yadan, 2019), Numpy (Harris et al., 2020), Scipy (Virtanen et al., 2020), Matplotlib (Hunter, 2007), and Pandas (McKinney et al., 2010). The code is publicly available at https://github.com/cambridge-mlg/neural_diffusion_processes.

F.1 Regression 1d

F.1.1 Data generation

We follow the same experimental setup as Bruinsma et al. (2020) to generate the 1d synthetic data. It consists of Gaussian (Squared Exponential (SE), MATÉRN($\frac{5}{2}$), WEAKLY PERIODIC) and non-Gaussian (SAWTOOTH and MIXTURE) sample paths, where MIXTURE is a combination of the other four datasets with equal weight. Fig. 9 shows samples for each of these dataset. The Gaussian datasets are corrupted with observation noise with variance $\sigma^2 = 0.05^2$. The left column of Fig. 9 shows example sample paths for each of the 5 datasets.

The training data consists of 2^{14} sample paths while the test dataset has 2^12 paths. For each test path we sample the number of context points between 1 and 10, the number of target points are fixed to 50 for the GP datasets and 100 for the non-Gaussian datasets. The input range for the training and interpolation datasets is $[-2, 2]$ for both the context and target sets, while for the extrapolation task the context and target input points are drawn from $[2, 6]$.

Architecture. For all datasets, except SAWTOOTH, we use 5 bi-dimensional attention layers (Dutordoir et al., 2022) with 64 hidden dimensions and 8 output heads. For SAWTOOTH, we obtained better performance with a wider and shallower model consisting of 2 bi-dimensional attention layers with a hidden dimensionality of 128. In all experiment, we train the NDP-based models over 300 epochs using a batch size of 256. Furthermore, we use the Adam optimiser for training with the following learning rate schedule: linear warm-up for 10 epochs followed by a cosine decay until the end of training.

F.1.2 Ablation Limiting Kernels

The test log-likelihoods (TLLs) reported in App. F.1.3 for the NDP models target a white limiting kernel and train to approximate the preconditioned score $K\nabla \log p_t$. Overall, we found this to be the best performing setting. App. F.1.3 shows an ablation study for different choices of limiting kernel and score parametrisation. We refer to Table 3 for a detailed derivation of the score parametrisations.

The dataset in the top row of the figure originates from a Squared Exponential (SE) GP with lengthscale $\ell = 0.25$. We compare the performance of three different limiting kernels: white (blue), a SE with a longer lengthscale $\ell = 1$ (orange), and a SE with a shorter lengthscale $\ell = 0.1$ (green). As the dataset is Gaussian, we have access to the true score. We observe that, across the different parameterisations, the white limiting kernel performance best. However, note that for the White kernel $K = I$ and thus the different parameterisations become identical. For non-white limiting kernels we see a reduction in performance for both the approximate and exact score. We attribute this to the additional complexity of learning a non-diagonal covariance.

In the bottom row of App. F.1.3 we repeat the experiment for a dataset consisting of samples from the Periodic GP with lengthscale 0.5. We draw similar conclusions: the best performing limiting kernel, across the different parametrisations, is the White noise kernel.

F.1.3 Ablation Conditional Sampling

Next, we focus on the empirical performance of the different noising schemes in the conditional sampling, as discussed in Fig. 8. For this, we measure the the Kullback-Leibler (KL) divergence between two Gaussian distributions: the true GP-based conditional distribution, and an distribution created by drawing conditional sampling from the model and fitting a Gaussian to it using the empirical mean and covariance. We perform this test on the 1D squared exponential dataset (described above) as this gives us access to the true posterior. We use 2^{12} samples to estimate the empirical mean and covariance, and fix the number of context points to 3.

In Fig. 11 we keep the total number of score evaluations fixed to 5000 and vary the number of steps in the inner (L) loop such that the number of outer steps is given by the ratio $5000/L$. From the figure, we observe that the particular choice of noising scheme is of less importance as long at least a couple (± 5) inner steps are taken. We further note that in this experiment we used the true score (available because of the Gaussianity of the dataset), which means that these results may differ if an approximate score network is used.

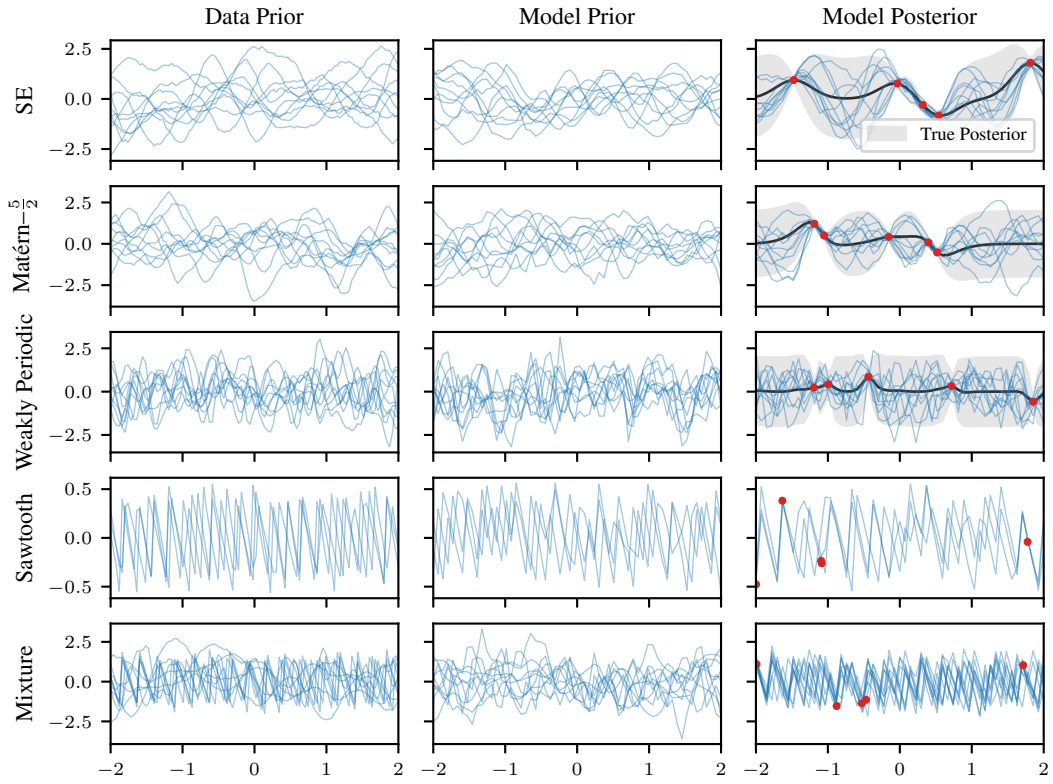
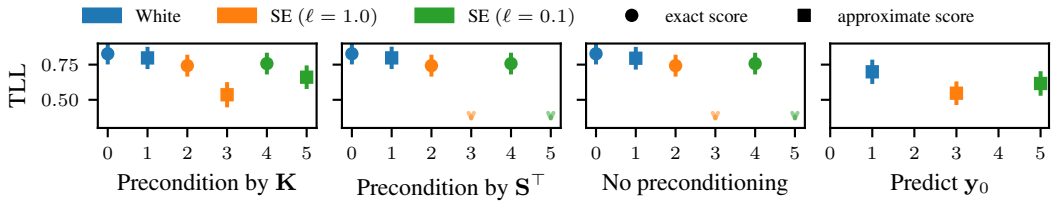
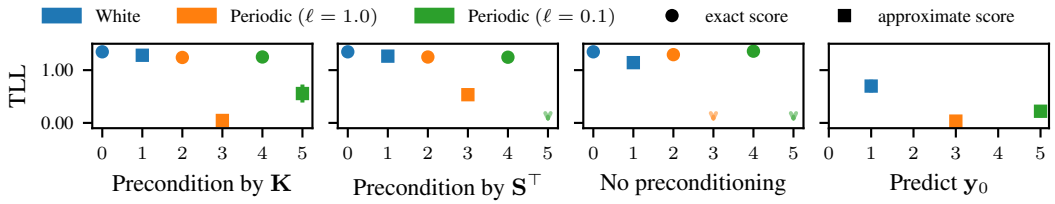


Figure 9: Visualisation of 1D regression experiment.



(a) Squared Exponential dataset with lengthscale $\ell = 0.25$



(b) Periodic dataset with lengthscale $\ell = 0.25$

Figure 10: *Ablation study* targeting different limiting kernels and score parametrisations.

Table 5: Mean test log-likelihood (TLL) $\pm$ 1 standard error estimated over 4096 test samples are reported. Statistically significant best non-GP model is in bold. The NP baselines (GNP, ConvCNP, ConvNP and ANP) are quoted from Bruinsma et al. (2020). '*' stands for a TLL below -10.

	SE	MATÉRN–$\frac{5}{2}$	WEAKLY PER.	SAWTOOTH	MIXTURE
INTERPOLATION					
GP (OPTIMUM)	0.70 ± 0.00	0.31 ± 0.00	-0.32 ± 0.00	n/a	n/a
$T(1)-$GEOMNDP	$\mathbf{0.72}\pm0.03$	$\mathbf{0.32}\pm0.03$	$\mathbf{-0.38}\pm0.03$	$\mathbf{3.39}\pm0.04$	$\mathbf{0.64}\pm0.08$
NDP*	$\mathbf{0.71}\pm0.03$	$\mathbf{0.30}\pm0.03$	$\mathbf{-0.37}\pm0.03$	$\mathbf{3.39}\pm0.04$	$\mathbf{0.64}\pm0.08$
GNP	$\mathbf{0.70}\pm0.01$	$\mathbf{0.30}\pm0.01$	-0.47 ± 0.01	0.42 ± 0.01	0.10 ± 0.02
CONVCNP	-0.80 ± 0.01	-0.95 ± 0.01	-1.20 ± 0.01	0.55 ± 0.02	-0.93 ± 0.02
CONVNP	-0.46 ± 0.01	-0.67 ± 0.01	-1.02 ± 0.01	1.20 ± 0.01	-0.50 ± 0.02
ANP	-0.61 ± 0.01	-0.75 ± 0.01	-1.19 ± 0.01	0.34 ± 0.01	-0.69 ± 0.02
GENERALISATION					
GP (OPTIMUM)	0.70 ± 0.00	0.31 ± 0.00	-0.32 ± 0.00	n/a	n/a
$T(1)-$GEOMNDP	$\mathbf{0.70}\pm0.02$	$\mathbf{0.31}\pm0.02$	$\mathbf{-0.38}\pm0.03$	$\mathbf{3.39}\pm0.03$	$\mathbf{0.62}\pm0.02$
NDP*	*	*	*	*	*
GNP	$\mathbf{0.69}\pm0.01$	$\mathbf{0.30}\pm0.01$	-0.47 ± 0.01	0.42 ± 0.01	0.10 ± 0.02
CONVCNP	-0.81 ± 0.01	-0.95 ± 0.01	-1.20 ± 0.01	0.53 ± 0.02	-0.96 ± 0.02
CONVNP	-0.46 ± 0.01	-0.67 ± 0.01	-1.02 ± 0.01	1.19 ± 0.01	-0.53 ± 0.02
ANP	-1.42 ± 0.01	-1.34 ± 0.01	-1.33 ± 0.00	-0.17 ± 0.00	-1.24 ± 0.01

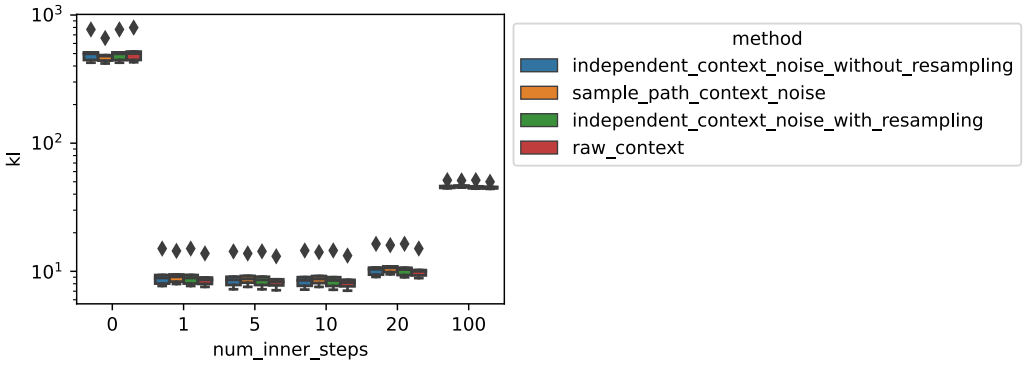


Figure 11: Ablation noising schemes for conditional sampling.

F.2 Gaussian process vector fields

Data We create synthetic datasets using samples from two-dimensional zero-mean GPs with the following E(2)-equivariant kernels: a diagonal Squared-Exponential (SE) kernel, a zero curl (CURL-FREE) kernel and a zero divergence (DIV-FREE) kernel, as described in App. D.1. We set the variance to $\sigma^2 = 1$ and the lengthscale to $\ell = \sqrt{5}$. We evaluate these GPs on a disk grid, created via a 2D grid with 30×30 points regularly space on $[-10, 10]^2$ and keeping only the points inside the disk of radius 10. We create a training dataset of size 80×10^3. and a test dataset of size 10×10^3.

Models We compare two flavours of our model GeomNDP. One with a non-equivariant attention-based score network (Figure C.1, Dutordoir et al., 2022), referred as NDP*. Another one with a E(2)-equivariant score architecture, based on steerable CNNs (Thomas et al., 2018; Weiler et al., 2018). We rely on the `e3nn` library (Geiger and Smidt, 2022) for implementation. A knn graph $\mathcal{E}$ is built with $k = 20$. The pairwise distances are first embed into $\mu(r_{ab})$ with a 'smooth_finite' basis of 50 elements via `e3nn.soft_one_hot_linspace`, and with a maximum radius of 2. The time is mapped via a sinusoidal embedding $\phi(t)$ (Vaswani et al., 2017). Then edge features are obtained as $e_{ab} = \Psi^{(e)}(\mu(r_{ab})||\phi(t)) \; \forall (a,b) \in \mathcal{E}_k$ with $\Psi^{(e)}$ an MLP with 2 hidden layers of width 64. We use 5 `e3nn.FullyConnectedTensorProduct` layers with update given by $V_a^{k+1} = \sum_{b \in \mathcal{N}(a,\mathcal{E}_k)} V_a^k \otimes \left(\Psi^v(e_{ab}||V_a^k||V_b^k)\right) Y(\hat{r}_{ab})$ with Y spherical harmonics up to order 2m Ψ^v an

MLP with 2 hidden layers of width 64 acting on invariant features, and node features V^k having irreps `12x0e + 12x0o + 4x1e + 4x1o`. Each layer has a gate non-linearity (Weiler et al., 2018).

We also evaluate two neural processes, a translation-equivariant CONVCNP (Gordon et al., 2020) with decoder architecture based on 2D convolutional layers (LeCun et al., 1998) and a $\mathrm{C}4 \ltimes \mathbb{R}^2 \subset \mathrm{E}(2)$-equivariant STEERCNP (Holderrieth et al., 2021) with decoder architecture based on 2D steerable convolutions (Weiler and Cesa, 2021). Specific details can be found in the accompanying codebase https://github.com/PeterHolderrieth/Steerable_CNPs of Holderrieth et al. (2021).

Optimisation. Models are trained for $80k$ iterations, via (Kingma and Ba, 2015) with a learning rate of $5e-4$ and a batch size of 32. The neural diffusion processes are trained unconditionally, that is we feed GP samples evaluated on the full disk grid. Their weights are updated via with exponential moving average, with coefficient 0.99. The diffusion coefficient is weighted by $\beta: \ t \mapsto \beta_{\min} + (\beta_{\max} - \beta_{\min}) \cdot t$, and $\beta_{\min} = 1e-4$, $\beta_{\max} = 15$.

As standard, the neural processes are trained by splitting the training batches into a context and evaluation set, similar to when evaluating the models. Models have been trained on A100-SXM-80GB GPUs.

Evaluation. We measure the predictive log-likelihood of the data process samples under the model on a held-out test dataset. The context sets are of size 25 and uniformly sampled from a disk grid of size 648, and the models are evaluated on the complementary of the grid. For neural diffusion processes, we estimate the likelihood by solving the associated probability flow ODE (53). The divergence is estimated with the Hutchinson estimator, with Rademacher noise, and 8 samples, whilst the ODE is solved with the 2nd order Heun solver, with 100 discretisation steps.

We also report the performance of the data-generating GP, and the same GP but with diagonal posterior covariance GP (DIAG.).

F.3 Tropical cyclone trajectory prediction

Data. The data is drawn from he International Best330 Track Archive for Climate Stewardship (IBTrACS) Project, Version 4 (Knapp et al., 2010; Knapp et al., 2018). The tracks are taken from the 'all' dataset covering the tracks from all cyclone basins across the globe. The tracks are logged at intervals of every 3 hours. From the dataset, we selected tracks of at least 50 time points long and clipped any longer to this length, resulting in 5224 cyclones. 90% was used fro training and 10% held out for evaluation. This split was changed across seeds. More interesting schemes of variable-length tracks or of interest, but not pursued here in this demonstrative experiment. Natively the track locations live in latitude-longitude coordinates, although it is processed into different forms for different models. The time stamps are processed into the number of days into the cyclone forming and this format is used commonly between all models.

Models.

Four models were evaluated.

The GP ($\mathbb{R} \to \mathbb{R}^2$) took the raw latitude-longitude data and normalised it. Using a 2-output RBF kernel with no covariance between the latitude and longitude and taking the cyclone time as input, placed a GP over the data. The hyperparameters of this kernel were optimised using a maximum likelihood grid search over the data. Note that this model places density outside the bounding box of $[-90, 90] \times [-180, 180]$ that defines the range of latitude and longitude, and so does not place a proper distribution on the space of paths on the sphere.

The STEREOGRAPHIC GP ($\mathbb{R} \to \mathbb{R}^2/\{0\}$) instead transformed the data under a sterographicc projection centred at the north pole, and used the same GP and optimisation as above. Since this model only places density on a set of measure zero that does not correspond to the sphere, it does induce a proper distribution on the space of paths on the sphere.

The NDP ($\mathbb{R} \to \mathbb{R}^2$) uses the same preprocessing as GP ($\mathbb{R} \to \mathbb{R}^2$) but uses a Neural Diffusion Process from (Dutordoir et al., 2022) to model the data. This has the same shortcomings as the GP ($\mathbb{R} \to \mathbb{R}^2$) in not placing a proper density on the space of paths on the sphere. The network used for the score function and the optimisation procedure is detailed below. A linear beta schedule was used with $\beta_0 = 1e-4$ and $\beta_1 = 10$. The reverse model was integrated back to $\epsilon = 5e-4$ for numerical

stability. The reference measure was a white noise kernel with a variance 0.05. ODEs and SDEs were discretised with 1000 steps.

The GEOMNDP ($\mathbb{R} \to \mathcal{S}^2$) works with the data projected into 3d space on the surface of the sphere. This projection makes no difference to the results of the model, but makes the computation of the manifold functions such as the exp map easier, and makes it easier to define a smooth score function on the sphere. This is done by outputting a vector for the score from the neural network in 3d space, and projecting it onto the tangent space of the sphere at the given point. For the necessity of this, see (De Bortoli et al., 2021). The network used for the score function and the optimisation procedure is detailed below. A linear beta schedule was used with $\beta_0 = 1e-4$ and $\beta_1 = 15$. The reverse model was integrated back to $\epsilon = 5e-4$ for numerical stability. The reference measure was a white noise kernel with a variance 0.05. ODEs and SDEs were discretised with 1000 steps.

Neural network. The network used to learn the score function for both NDP ($\mathbb{R} \to \mathbb{R}^2$) and GEOMNDP ($\mathbb{R} \to \mathcal{S}^2$) is a bi-attention network from Dutordoir et al. (2022) with 5 layers, hidden size of 128 and 4 heads per layer. This results in 924k parameters.

Optimisation. NDP ($\mathbb{R} \to \mathbb{R}^2$) and GEOMNDP ($\mathbb{R} \to \mathcal{S}^2$) were both optimised using (correctly implemented) Adam for 250k steps using a batch size of 1024 and global norm clipping of 1. Batches were drawn from the shuffled data and refreshed each time the dataset was exhausted. A learning rate schedule was used with 1000 warmup steps linearly from 1e-5 to 1e-3, and from there a cosine schedule decaying from 1e-3 to 1e-5. With even probability either the whole cyclone track was used in the batch, or 20 random points were sub-sampled to train the model better for the conditional sampling task.

Conditional sampling. The GP models used closed-form conditional sampling as described. Both diffusion-based models used the Langevin sampling scheme described in this work. 1000 outer steps were used with 25 inner steps. We use a $\psi = 1.0$ and $\lambda_0 = 2.5$. In addition at the end of the Langevin sampling, we run an additional 150 Langevin steps with $t = \epsilon$ as this visually improved performance.

Evaluation. For the model (conditional) log probabilities the GP models were computed in closed form. For the diffusion-based models, they were computed using the auxiliary likelihood ODE discretised over 1000 steps. The conditional probabilities were computed via the difference between the log-likelihood of the whole trajectory and the log-likelihood of the context set only. The mean squared errors were computed using the geodesic distance between 10 conditionally sampled trajectories, described above.