

# SPACETIME GEOMETRY OF DENOISING IN DIFFUSION MODELS

A PREPRINT

**Rafał Karczewski**  
Aalto University  
rafal.karczewski@aalto.fi

**Markus Heinonen**  
Aalto University  
markus.o.heinonen@aalto.fi

**Alison Pouplin**  
Aalto University  
alison.pouplin@aalto.fi

**Søren Hauberg**  
Technical University of Denmark  
sohau@dtu.dk

**Vikas Garg**  
Aalto University  
YaiYai Ltd  
vgarg@csail.mit.edu

## ABSTRACT

We present a novel perspective on diffusion models using the framework of information geometry. We show that the set of noisy samples, taken across all noise levels simultaneously, forms a statistical manifold – a family of denoising probability distributions. Interpreting the noise level as a temporal parameter, we refer to this manifold as *spacetime*. This manifold naturally carries a Fisher-Rao metric, which defines geodesics – shortest paths between noisy points. Notably, this family of distributions is exponential, enabling efficient geodesic computation even in high-dimensional settings without retraining or fine-tuning. We demonstrate the practical value of this geometric viewpoint in transition path sampling, where spacetime geodesics define smooth sequences of Boltzmann distributions, enabling the generation of continuous trajectories between low-energy metastable states. Code is available at: <https://github.com/Aalto-QuML/diffusion-spacetime-geometry>.

## 1 Introduction

Diffusion models have emerged as a powerful paradigm for generative modeling, demonstrating remarkable success in learning to model and sample data (Yang et al., 2023). Diffusion models gradually corrupt a data sample  $x_0 \in \mathbb{R}^D$  into Gaussian noise  $x_T$  through  $T$  forward steps, then learn to reverse this corruption to recover the original sample. While the underlying mathematical frameworks of training and sampling are well-established (Sohl-Dickstein et al., 2015; Kingma et al., 2021; Song et al., 2021b; Lu et al., 2022; Holderrieth et al., 2025), analysing how information evolves through the noisy intermediate states  $x_t$  where  $t \in [0, T]$  remains an open question.

We ask: Can we equip the diffusion’s latent space with a tractable geometric structure?

To address this, we must first define what *latent space* means for diffusion models, a concept that lacks a universally accepted definition. Typically, the final noise vector  $x_T$  is regarded as the latent representation, with the denoiser (e.g. PF-ODE) being the decoder  $x_0(x_T)$  (Song et al., 2021b). However, this treats the PF-ODE as a black box and overlooks the intrinsic spatio-temporal structure of diffusion models. Notably, intermediate noisy states have already proven useful in defence against adversarial attacks (Yoon et al., 2021), or image editing (Park et al., 2023). Thus, we propose to define the latent space as the entire *spacetime* continuum  $(x_t, t)$  for  $t \in (0, T]$ .

Similarly, we need to clarify the notion of *geometry*. The most commonly adopted approach to defining geometry in the latent space is via the pullback metric (Arvanitidis et al., 2018, 2022), which is given by the Jacobian of the decoder

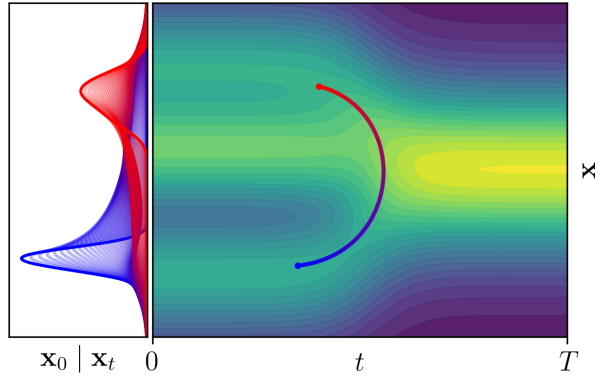


Figure 1: **A geodesic in spacetime** is the shortest path between denoising distributions.

and describes how small changes in the latent code affect the decoding. However, we will show that the decoder  $\mathbf{x}_0(\mathbf{x}_t)$ , defined as the solution of an ODE, makes evaluating geometric quantities computationally prohibitive.

Surprisingly, this computational burden can be lifted by shifting focus from the deterministic decoder  $\mathbf{x}_0(\mathbf{x}_t)$  to the denoising distribution  $p(\mathbf{x}_0|\mathbf{x}_t)$ . This reframing enables the use of the Fisher-Rao metric (Amari, 2016), which naturally defines the geometry on a continuous space of distributions. Importantly, we show that the denoising distributions  $\mathbf{x}_0|\mathbf{x}_t$  form an exponential family, which enables, surprisingly, tractable estimation of *geodesics* in the spacetime  $(\mathbf{x}_t, t)$  (See Fig. 1).

We summarize our contributions below. In this work, we

- study the latent space of diffusion models as a  $(D + 1)$ -dimensional statistical manifold,
- derive its Fisher-Rao metric characterizing the geometry underlying the denoising process,
- demonstrate that the geodesics between any two samples become tractable due to a novel insight that the denoising distributions form an exponential family,
- show transition paths in image-based denoisers, and in molecular applications.

## 2 Background

### 2.1 Diffusion models

We assume a data distribution  $q$  defined on  $\mathbb{R}^D$ , and the forward process

$$p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I}), \quad (1)$$

which gradually transforms  $q$  into pure noise  $p_T \approx \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$  at time  $T$ , where  $\alpha_t, \sigma_t$  are hyperparameters such that  $\text{SNR}(t) = \alpha_t^2/\sigma_t^2$  is decreasing. This process is equivalent to a Stochastic Differential Equation (SDE) (Song et al., 2021b)

$$\text{Forward SDE: } d\mathbf{x} = f(t)\mathbf{x}dt + g(t)d\mathbf{W}_t, \quad \mathbf{x}_0 \sim q, \quad (2)$$

where  $f(t) = \frac{d}{dt} \log \alpha_t$ ,  $g^2(t) = -\sigma_t^2 \frac{d\lambda_t}{dt}$  for  $\lambda_t = \log \text{SNR}(t)$ , and  $\mathbf{W}$  is the Wiener process. There exists a corresponding *denoising* SDE, which reverses this process (Anderson, 1982)

$$\text{Reverse SDE: } d\mathbf{x} = \left( f(t)\mathbf{x} - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right) dt + g(t)d\bar{\mathbf{W}}_t, \quad \mathbf{x}_T \sim p_T, \quad (3)$$

where  $p_t$  is the marginal distribution of the forward process (Eq. 2) at time  $t$ , and  $\bar{\mathbf{W}}$  the reverse-time Wiener process. Somewhat unexpectedly, there exists a deterministic Probability Flow Ordinary Differential Equation (PF-ODE), which shares marginal distributions with both SDEs (Song et al., 2021b):

$$\text{PF ODE: } d\mathbf{x} = \left( f(t)\mathbf{x} - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right) dt, \quad \mathbf{x}_T \sim p_T. \quad (4)$$

### 2.2 Statistical manifolds

Information geometry is a subfield of Riemannian geometry studying *statistical manifolds*, i.e., families of distributions  $\mathcal{P} = \{p(\cdot|\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$  parameterised by  $\boldsymbol{\theta}$  (Amari, 2016) (See Fig. 2). For an introduction to information geometry, we refer to Nielsen (2020) or Mishra et al. (2023).

A central task in information geometry is finding shortest paths between two distributions  $p(\cdot|\boldsymbol{\theta})$  and  $p(\cdot|\boldsymbol{\theta}')$  along the manifold  $\mathcal{P}$ . To measure how small changes in  $\boldsymbol{\theta}$  affect the distribution  $p(\cdot|\boldsymbol{\theta})$  we use the *Fisher-Rao* metric tensor

$$\mathcal{I}_{\boldsymbol{\theta}} = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})} \left[ \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta})^\top \right] \in \mathbb{R}^{\dim(\Theta) \times \dim(\Theta)}, \quad (5)$$

which locally approximates the Kullback-Leibler divergence (Amari, 2016)

$$\text{KL} \left[ p(\cdot|\boldsymbol{\theta}) \parallel p(\cdot|\boldsymbol{\theta} + d\boldsymbol{\theta}) \right] = \frac{1}{2} d\boldsymbol{\theta}^\top \mathcal{I}_{\boldsymbol{\theta}} d\boldsymbol{\theta} + o(\|d\boldsymbol{\theta}\|^2). \quad (6)$$

Note that  $\mathcal{I}_{\boldsymbol{\theta}}$  coincides with the Fisher information matrix (FIM) in statistics (Ly et al., 2017).

Up to scale, the Fisher-Rao metric is the only Riemannian metric on the space of probability distributions that is invariant under sufficient statistics (Cencov, 1981). In other words, it is the only metric that treats statistically equivalent representations of data as geometrically equivalent.

### 2.3 Geodesics

The metric (Eq. 5) enables defining length of curves  $\gamma : [0, 1] \rightarrow \Theta$  of distributions between endpoints  $\theta_0$  and  $\theta_1$  as

$$\ell(\gamma) = \int_0^1 \|\dot{\gamma}_s\|_{\mathcal{I}} ds = \int_0^1 \sqrt{\dot{\gamma}_s^\top \mathcal{I}_{\gamma(s)} \dot{\gamma}_s} ds. \quad (7)$$

The distance between  $\theta_0$  and  $\theta_1$  is defined as the length of the shortest curve connecting them,

$$d_{\mathcal{I}}(\theta_0, \theta_1) = \inf \left\{ \ell(\gamma) \mid \gamma_0 = \theta_0, \gamma_1 = \theta_1 \right\}, \quad (8)$$

and  $\gamma$  realizing this distance is called the *geodesic*. This is also the curve minimizing the energy (Do Carmo and Francis, 1992):

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_0^1 \|\dot{\gamma}_s\|_{\mathcal{I}}^2 ds. \quad (9)$$

Geodesics, thus, naturally characterize the simplest transition path between two distributions, and their lengths induce a distance measure on  $\mathcal{P}$ .

### 2.4 Exponential Family

In the case when the distribution family is exponential, the Fisher-Rao metric simplifies considerably. A family of distributions  $\mathcal{P} = \{p(\cdot|\theta) \mid \theta \in \Theta\}$  is called *exponential* when

$$p(x|\theta) = h(x) \exp \left( \eta(\theta)^\top T(x) - \psi(\theta) \right), \quad (10)$$

where  $T(x)$  represents the *sufficient statistic*, a function of  $x$  that captures all the information about  $\theta$ , and  $\eta(\theta)$  the *natural parameter*.  $h$  and  $\psi$  are respectively a base measure and a log-partition function ensuring correct normalization of the probability density. We derive a modified version of the Fisher-Rao metric for exponential families (Nielsen and Garcia, 2009), which is more suitable for our purposes (cf. Appendix B.1):

$$\mathcal{I}_\theta = \left( \frac{\partial \eta(\theta)}{\partial \theta} \right)^\top \left( \frac{\partial \mu(\theta)}{\partial \theta} \right), \quad (11)$$

where

$$\mu(\theta) = \mathbb{E}[T(x)|\theta] = \int T(x) p(x|\theta) dx \quad (12)$$

is the *expectation parameter*. The Fisher-Rao metric (Eq. 11) for exponential families captures how changes in  $\theta$  affect both the natural parameter  $\eta$  and the expectation parameter  $\mu$ . The squared norm of a direction  $v$  under this metric is the dot product of the directional derivatives  $D_v \eta$  and  $D_v \mu$ , measuring how aligned the changes in  $\eta$  and  $\mu$  are under small shifts in  $\theta$ .

The Fisher-Rao metric  $\mathcal{I}$  applied to exponential families (Eq. 11) translates into a simpler expression for the geodesic energy (Eq. 9) (cf. Appendix B.2):

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_0^1 \left( \frac{d}{ds} \eta(\gamma_s) \right)^\top \left( \frac{d}{ds} \mu(\gamma_s) \right) ds. \quad (13)$$

That is, all relevant information about  $\gamma$  is encapsulated within  $\eta$  and  $\mu$ , and remarkably, the energy is independent of  $h$  and  $\psi$ , which are typically difficult to estimate. As we will show later, this insight leads to an efficient algorithm for computing geodesics.

## 3 Denoising spacetime geometry

Considering all latent representations  $x_t$  of data points  $x_0 \in \mathbb{R}^D$  across all noise levels  $t \in (0, T]$ , we obtain a latent space structured as the set of pairs  $(x_t, t) \in \mathbb{R}^D \times (0, T]$ , forming a  $(D + 1)$ -dimensional manifold. The PF-ODE,  $x_0^{\text{PF}} : \mathcal{X} \times (0, T] \rightarrow \mathcal{X}$  (Eq. 4), then acts as a decoder mapping these  $(x_t, t)$  pairs back to the data space. The standard

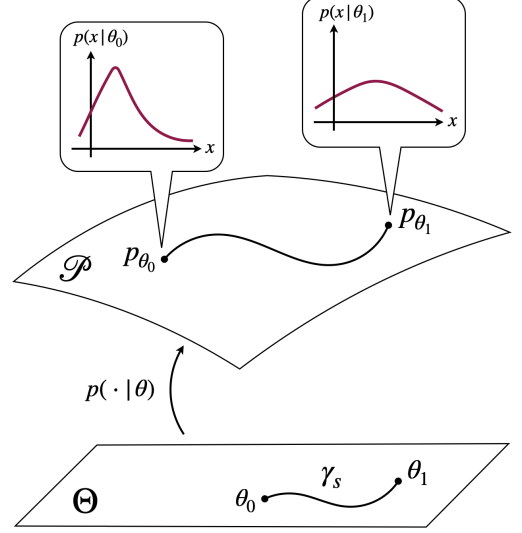


Figure 2: **Statistical manifold:** a continuum of distributions  $p(\cdot|\theta)$  with geodesics on  $\Theta$  enabling smooth transitions.

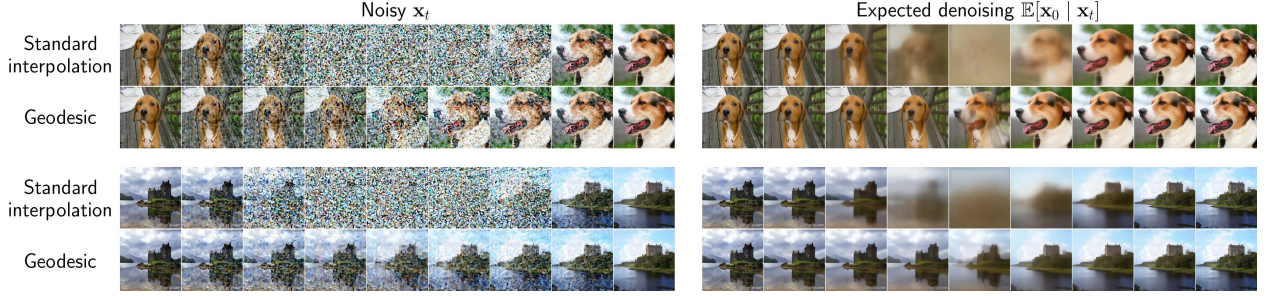


Figure 3: **Spacetime geodesics between clean images remove less information than interpolation through full noise** (Song et al., 2021a,b). Left: spacetime curve. Right: denoising mean.

approach to defining a geometric structure in the latent space  $(\mathbf{x}_t, t)$  is via the pullback metric (Arvanitidis et al., 2018, 2022)

$$\text{PF-ODE pullback metric: } M_{(\mathbf{x}_t, t)} = \frac{\partial \mathbf{x}_0^{\text{PF}}(\mathbf{x}_t, t)}{\partial (\mathbf{x}_t, t)}^\top \frac{\partial \mathbf{x}_0^{\text{PF}}(\mathbf{x}_t, t)}{\partial (\mathbf{x}_t, t)} \in \mathbb{R}^{(D+1) \times (D+1)}. \quad (14)$$

However, in the case of diffusion models, the decoder being a solution of an ODE introduces significant computational challenges (See Appendix C).

We show that adopting a stochastic perspective, by considering the denoising distribution  $p(\mathbf{x}_0 | \mathbf{x}_t)$  as a stochastic decoder, leads to significant computational savings. Rather than using the standard pullback geometry, which analyzes how small changes in latent noisy points affect the decoded sample, we take an information geometric view: we study how such changes affect the entire denoising distribution. Crucially, this can be evaluated without any costly ODE or SDE simulations required in the pullback framework.

### 3.1 Spacetime as a statistical manifold

In diffusion models, a sample  $\mathbf{x}_t \in \mathbb{R}^D$  at noise level  $t \in (0, T]$  arises from corrupting a clean signal  $\mathbf{x}_0$  via a forward process (Eq. 1). The associated *denoising distribution*,

$$p(\mathbf{x}_0 | \mathbf{x}_t) \propto p(\mathbf{x}_t | \mathbf{x}_0) q(\mathbf{x}_0), \quad (15)$$

describes the distribution over possible clean samples  $\mathbf{x}_0$  that could have produced the noisy observation  $\mathbf{x}_t$  at time  $t$ .<sup>1</sup> As we vary  $(\mathbf{x}_t, t)$ , we obtain a family of denoising distributions

$$\mathcal{P} = \left\{ p(\mathbf{x}_0 | \mathbf{x}_t) \mid \mathbf{x}_t \in \mathbb{R}^D, t \in (0, T] \right\}, \quad (16)$$

which defines a statistical manifold with parameters  $\boldsymbol{\theta} = (\mathbf{x}_t, t) \in \Theta \subseteq \mathbb{R}^D \times (0, T]$ . Each point on this manifold corresponds to a distinct denoising distribution.

### 3.2 Denoising manifold is an exponential family

Although the denoising distributions  $p(\mathbf{x}_0 | \mathbf{x}_t)$  in diffusion models are generally intractable and difficult to approximate (Rissanen et al., 2025), we show that they take the form of an exponential family, with explicitly defined parameters. We show that both the natural and expectation parameters have closed-form expressions. Moreover, the expectation parameter corresponds precisely to the first and second denoising moments, which directly lead to a computationally efficient approximation of the curve energy.

**Proposition 1** (Exponential family of denoising). *Let  $\mathbf{x}_t$  be a noisy observation corresponding to diffusion time  $t$ , as introduced in Eq. 1. Then*

$$p(\mathbf{x}_0 | \mathbf{x}_t) = h(\mathbf{x}_0) \exp \left( \boldsymbol{\eta}(\mathbf{x}_t, t)^\top T(\mathbf{x}_0) - \psi(\mathbf{x}_t, t) \right), \quad (17)$$

<sup>1</sup>Note that  $p(\mathbf{x}_0 | \mathbf{x}_t) = p(\mathbf{x}_0 | \mathbf{x}_t, t)$  also implicitly depends on  $t$ , which we omit in the notation.

with  $h = q$  the data distribution density,  $\psi$  the log-partition function, and

$$\boldsymbol{\eta}(\mathbf{x}_t, t) = \left( \frac{\alpha_t}{\sigma_t^2} \mathbf{x}_t, -\frac{\alpha_t^2}{2\sigma_t^2} \right) \quad (\text{natural parameter}) \quad (18)$$

$$T(\mathbf{x}_0) = (\mathbf{x}_0, \|\mathbf{x}_0\|^2) \quad (\text{sufficient statistic}) \quad (19)$$

$$\boldsymbol{\mu}(\mathbf{x}_t, t) = \left( \underbrace{\frac{1}{\alpha_t} \left( \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \right)}_{\text{'space': } \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]}, \underbrace{\frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2}_{\text{'time': } \mathbb{E}[\|\mathbf{x}_0\|^2 | \mathbf{x}_t]} \right) \quad (20)$$

For the complete proof, see [Appendix D](#).

### 3.3 Denoiser network approximation

In practice, we approximate the denoising mean  $\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]$  with a neural network denoiser

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) \approx \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t], \quad (21)$$

which leads to the approximation of the expectation parameter:

$$\boldsymbol{\mu}(\mathbf{x}_t, t) \approx \left( \hat{\mathbf{x}}_0(\mathbf{x}_t), \frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \hat{\mathbf{x}}_0(\mathbf{x}_t) + \|\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2 \right) \in \mathbb{R}^{D+1}. \quad (22)$$

To estimate  $\text{div}_{\mathbf{x}_t} \hat{\mathbf{x}}_0(\mathbf{x}_t)$  efficiently, we apply the Hutchinson’s trick ([Hutchinson, 1989](#)) with a single Rademacher variable ([Grathwohl et al., 2019](#)), reducing the cost of estimating  $\boldsymbol{\mu}$  to a single Jacobian-vector-product (JVP) (more details in [Appendix G](#)).

We approximate the energy  $\mathcal{E}(\boldsymbol{\gamma})$  ([Eq. 13](#)) via numerical integration and finite differences of  $\dot{\boldsymbol{\eta}}, \dot{\boldsymbol{\mu}}$

$$\mathcal{E}(\boldsymbol{\gamma}) \approx \frac{1}{2ds} \sum_{n=0}^{N-2} \left( \boldsymbol{\eta}(\boldsymbol{\gamma}_{n+1}) - \boldsymbol{\eta}(\boldsymbol{\gamma}_n) \right)^\top \left( \boldsymbol{\mu}(\boldsymbol{\gamma}_{n+1}) - \boldsymbol{\mu}(\boldsymbol{\gamma}_n) \right), \quad (23)$$

on the discretized curve into  $N$  points  $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_0, \dots, \boldsymbol{\gamma}_{N-1})$ . Evaluating the energy requires  $N$  JVP evaluations of the denoiser network. We optimise  $\boldsymbol{\gamma}$  by minimizing  $\mathcal{E}(\boldsymbol{\gamma})$  using gradient descent.

We compare the energy estimation cost with the standard pullback metric approach:

$$\begin{aligned} \text{Pullback geometry: } & \mathcal{O}(NK) \text{ evaluations of } \hat{\mathbf{x}}_0(\mathbf{x}_t) \\ \text{Information geometry: } & \mathcal{O}(N) \text{ JVPs of } \hat{\mathbf{x}}_0(\mathbf{x}_t) \end{aligned}$$

with  $K$  the number of solver steps. Since the JVP costs about twice a denoiser evaluation, and  $K \gg 2$ , the information geometry approach is much more efficient when optimizing the energy to find geodesics (see [Appendix C](#) for details).

### 3.4 Interpolating between data through spacetime

As a demonstration of our geometric framework, we perform image interpolations. A common approach to interpolating between  $\mathbf{x}_0^a$  and  $\mathbf{x}_0^b$  is encoding both points with PF-ODE from  $t = 0$  to  $t = T$ , and connecting them with spherical linear interpolation (SLERP) ([Song et al., 2021a,b](#)):

$$\mathbf{x}_0^a \xrightarrow{\text{PF-ODE}} \mathbf{x}_T^a \xrightarrow{\text{SLERP}} \mathbf{x}_T^b \xrightarrow{\text{PF-ODE}} \mathbf{x}_0^b. \quad (24)$$

This interpolation path is a curve in spacetime, but not the shortest one. We compare this interpolation path with a geodesic in [Fig. 3](#) (See [Appendix E.2](#) for implementation details). We note that the standard approach to interpolation, by encoding the image to  $t = T$ , removes all information before generating it again. In contrast, the geodesic minimizes information loss, discarding only what is essential to move between images.

After decoding the noisy intermediate images on the interpolation path with PF-ODE, we found that the spacetime geodesic

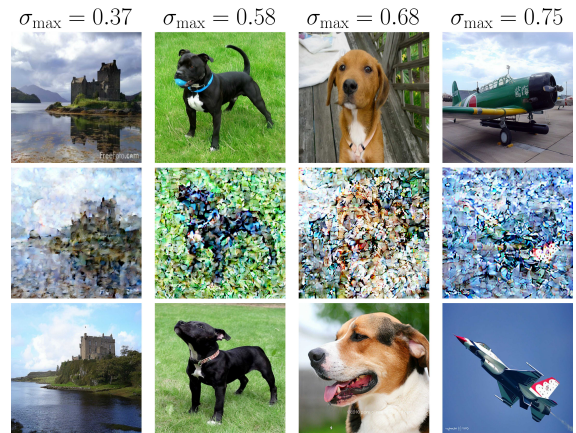


Figure 4: **Peak noise on the geodesic depends on endpoint similarity.** Top/bottom: endpoints. Middle: geodesic point at peak noise.



introduces less semantic changes than the standard approach, however the images decoded from the geodesic were less realistic (blurry), which we hypothesize might stem from the fact that the approximate geodesic minimization did not find the optimal path. Please see Fig. 8 for visualization.

Furthermore, we observe that the amount of added noise on the interpolating geodesic depends on how similar the endpoints are. If the starting two images are similar, less noise is needed than for dissimilar images. See Fig. 4 for a few examples and Appendix E.2 for details.

## 4 Experiments

### 4.1 Sampling trajectories

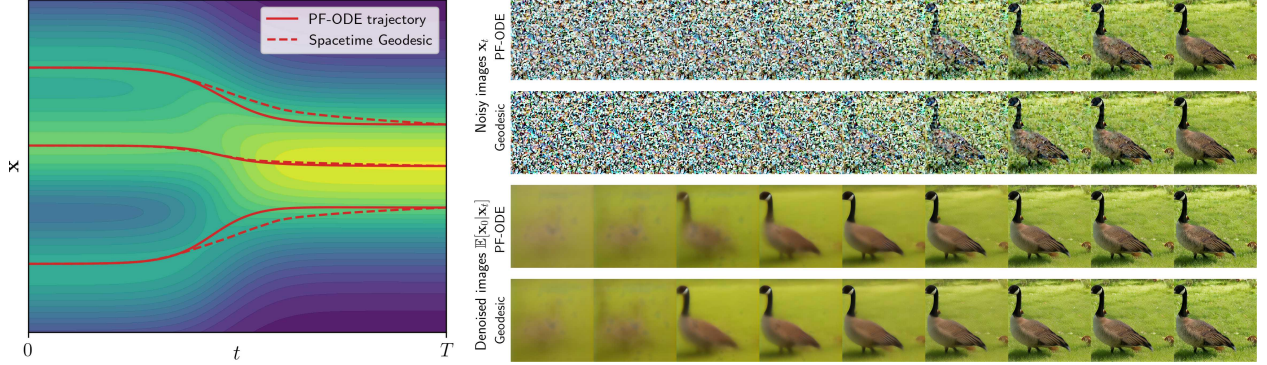


Figure 5: **PF-ODE paths are similar to energy-minimizing geodesics.** Left: Geodesics move in straighter lines than PF-ODE trajectories in 1D toy density. Right: Geodesics are almost indistinguishable to PF-ODE sampling in ImageNet-512 EDM2 model.

We compare the trajectories obtained by solving the PF-ODE  $x_0(x_T)$  (Eq. 4) with geodesics (Eq. 23) between the same endpoints  $x_0, x_T$ . For a toy example of 1D mixture of Gaussians, we observe the geodesics curving less than the PF-ODE trajectories in the early sampling (high  $t$ ), while being indistinguishable for lower values of  $t$  (See Fig. 5 left and Appendix E.1 for details).

We find only marginal perceptual difference between the PF-ODE sampling trajectories and the geodesics in the EDM2 ImageNet-512 model (Karras et al., 2024). The geodesic appears to generate information slightly earlier, but the difference is minor (See Fig. 5 right, and Appendix E.2 for details).

We note that spacetime geodesics are not an alternative sampling method since they require knowing the endpoints beforehand. An investigation into whether our framework can be used to improve sampling strategies is an interesting future research direction.

### 4.2 Transition path sampling

We consider the problem of transition-path sampling (Holdijk et al., 2023; Du et al., 2024; Raja et al., 2025), whose goal is to find probable transition paths between low-energy states. We assume a Boltzmann distribution

$$q(\mathbf{x}) \propto \exp(-U(\mathbf{x})), \quad (25)$$

where  $U$  is a known energy function, which is a common assumption in molecular dynamics.

In this setting, the denoising distribution follows a tractable energy function (See Eq. 58)

$$p(x_0|x_t) \propto q(x_0)p(x_t|x_0) \propto \exp \left( \underbrace{-U(x_0) - \frac{1}{2}\text{SNR}(t)\|x_0 - x_t/\alpha_t\|^2}_{-U(x_0|x_t)} \right). \quad (26)$$

To construct a transition path between two low-energy states  $x_0^1$  and  $x_0^2$ , we estimate the spacetime geodesic  $\gamma$  between them using a denoiser model  $\hat{x}_0(x_t) \approx \mathbb{E}[x_0|x_t]$  with Eq. 23, as shown in Fig. 6. At each interpolation point  $s \in [0, 1]$ , the geodesic defines a denoising Boltzmann distribution  $p(x|\gamma_s)$  where  $U(x|\gamma_s)$  is the energy at that spacetime location. See Appendix E.3 for details.

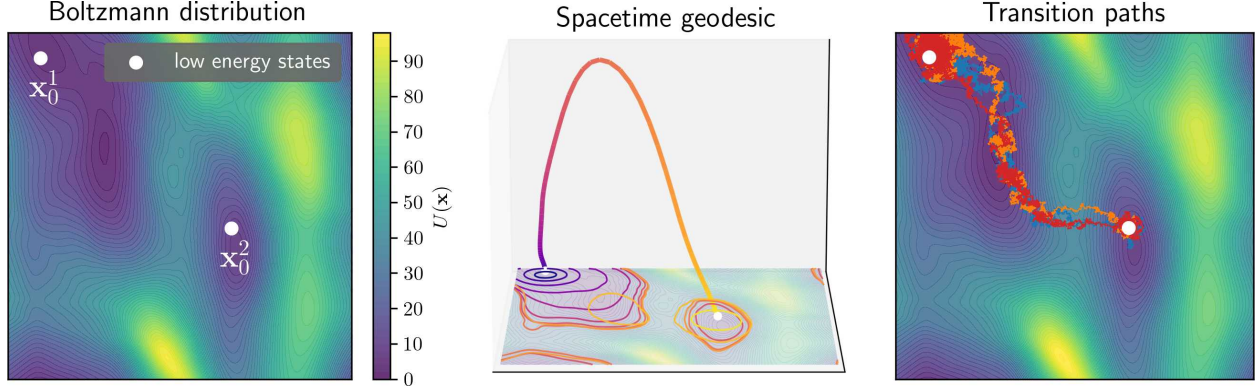


Figure 6: **Spacetime geodesics enable sampling transition paths between low-energy states.** Left: Alanine Dipeptide energy landscape wrt two dihedral angles, with two energy minima  $x_0^1, x_0^2$ . Middle: Spacetime geodesic  $\gamma$  connecting  $x_0^1$  and  $x_0^2$ . Right: Annealed Langevin transition path samples.

**Annealed Langevin Dynamics** To sample transition paths we use Langevin dynamics

$$d\mathbf{x} = -\nabla_{\mathbf{x}} U(\mathbf{x}|\gamma_s)dt + \sqrt{2}d\mathbf{W}_t, \quad (27)$$

whose stationary distributions are  $p(\mathbf{x} | \gamma_s) \propto \exp(-U(\mathbf{x}|\gamma_s))$  for any  $s$ . To obtain the trajectories from  $x_0^1$  to  $x_0^2$ , we gradually increase  $s$  from 0 to 1 using annealed Langevin (Song and Ermon, 2019). After discretizing the geodesic into  $N$  points  $\gamma_n$ , we alternate between taking  $K$  steps of Eq. 27 conditioned on  $\gamma_n$  and updating  $\gamma_n \mapsto \gamma_{n+1}$ , as described in Algorithm 1. This approach assumes that  $p(\mathbf{x}|\gamma_n)$  is close to  $p(\mathbf{x}|\gamma_{n+1})$ , and thus  $\mathbf{x} \sim p(\mathbf{x}|\gamma_n)$  is a good starting point to Langevin Dynamics conditioned on  $\gamma_{n+1}$ .

**Example** We compute a spacetime geodesic connecting two molecular configurations of Alanine Dipeptide, as in Holdijk et al. (2023). In Fig. 6, the energy landscape is visualized over the dihedral angle space, with a neural network used to approximate the potential energy  $U$ . Using our trained denoiser  $\hat{x}_0(x_t)$ , we estimate the expectation parameter  $\mu$ , which allows us to compute and visualize a geodesic trajectory through spacetime. Transition paths were sampled via annealed Langevin dynamics. See Appendix E.3 for details.

---

**Algorithm 1** Transition Path Sampling with Annealed Langevin Dynamics

---

**Require:**  $\mathbf{x}_a, \mathbf{x}_b \in \mathbb{R}^D$  endpoints,  $N_\gamma > 0, T > 0, t_{\min}, dt$

- 1:  $\gamma \leftarrow \arg \min_{\gamma} \mathcal{E}(\gamma)$  ▷ Approximate spacetime geodesic connecting  $\mathbf{x}_a$  with  $\mathbf{x}_b$
- 2:  $\mathcal{T} \leftarrow \{\mathbf{x} := \mathbf{x}_a\}$  ▷ Initialize chain  $\mathcal{T}$  at  $\mathbf{x}_a$
- 3: **for**  $n \in \{0, \dots, N_\gamma - 1\}$  **do** ▷ Iterate over the points on the geodesic  $\gamma_n$
- 4:   **for**  $t \in \{1, \dots, T\}$  **do**
- 5:      $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ▷ Sample Gaussian noise
- 6:      $\mathbf{x} \leftarrow \mathbf{x} - \nabla_{\mathbf{x}} U(\mathbf{x}|\gamma_n)dt + \sqrt{2dt}\varepsilon$  ▷ Langevin update
- 7:      $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{x}\}$  ▷ Append state  $\mathbf{x}$  to chain
- 8:   **end for**
- 9: **end for**
- 10: **return**  $\mathcal{T}$  ▷ Return chain

---

### 4.3 Constrained path sampling

Suppose we would like to impose additional constraints along the geodesic interpolants. This corresponds to constrained optimization

$$\min_{\gamma} \left\{ \mathcal{E}(\gamma) + \lambda \int_0^1 h(\gamma_s) ds, \quad \text{s.t.} \quad \gamma_0 = (\mathbf{x}_0^1, 0), \gamma_1 = (\mathbf{x}_0^2, 0) \right\}, \quad (28)$$

where  $h: \mathbb{R} \times \mathbb{R}^D \rightarrow \mathbb{R}$  is some penalty function with  $\lambda > 0$ . We demonstrate the principle by (i) penalising transition path variance, and (ii) imposing regions to avoid in the data space.

**Low variance transitions** Suppose we want  $p(x|\gamma_s)$ 's to have small variance. Equation (Eq. 62) in Appendix D shows that high SNR implies low denoising variance. Therefore, we can reduce variance by choosing  $h(x_t) = \max(-\log \text{SNR}(t), \rho)$  for some threshold  $\rho$ .

**Avoiding restricted regions** Suppose we want to avoid certain regions in the data space in the transition paths. We encode the region to avoid as a denoising distribution  $p(\cdot|\theta^*)$  for some  $\theta^* = (x_t^*, t^*)$  where larger the  $t^*$ , larger the restricted region. We encode the penalty as KL distance between the denoising distributions (See Appendix B.3)

$$\text{KL} [p(\cdot|\theta^*)||p(\cdot|\gamma_s)] = \int_0^s \left( \frac{d}{du} \eta(\gamma_u) \right)^\top (\mu(\gamma_u) - \mu(\theta^*)) du + C \quad (29)$$

$$h(\gamma_s) = \min \left( \rho, -\text{KL} [p(\cdot|\theta^*)||p(\cdot|\gamma_s)] \right). \quad (30)$$

**Example** Fig. 7 shows an example where we define  $\theta^* = (-0.8, -0.1, t^*)$  with  $\log \text{SNR}(t^*) = 4$ . The figure shows the transition paths successfully avoiding the penalty region  $p(\cdot|\theta^*)$ , where the restricted region denotes where  $U(x|\theta^*) - \min_x U(x|\theta^*) = 15$ . See Appendix E.3 for experiment details.

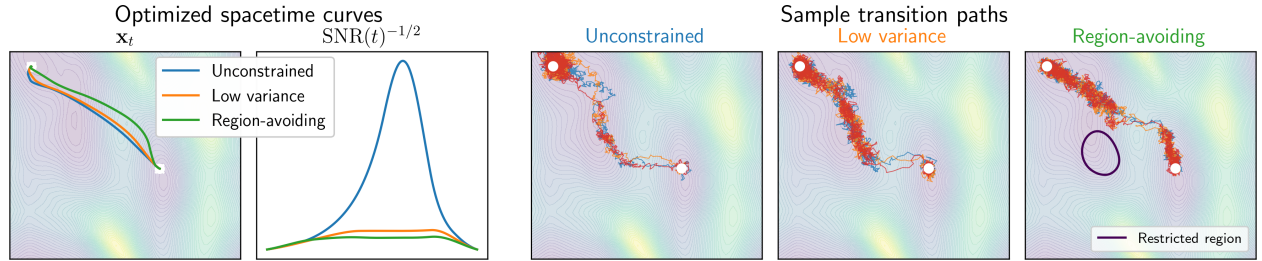


Figure 7: **Vanilla transition paths can be constrained to have lower variance, or successfully avoid a restricted region  $p(\cdot|\theta^*)$ .** Left: geodesics  $\gamma$ . Right: transition paths  $T$ .

## 5 Related works

Previous studies have separately examined the impact of latent noise on data and employed geometric frameworks to extend or analyze diffusion models. We review those previous contributions to provide a broader context.

**Latent structures** Prior work has examined the relationship between latent noise  $x_t$  and data  $x_0$  in score-based models. Yu et al. (2025) propose a geodesic density in diffusion latent space, Park et al. (2023) apply Riemannian geometry to lower-dimensional latent code, and Karczewski et al. (2025) study how scaling noise affects the log-density and perceptual detail of generated images. Our research also studies the  $x_t$  to  $x_0$  relationship but differs in three ways: we use the principled Fisher–Rao metric rather than a less motivated inverse-density metric, maintain the full-dimensional latent space without projection, and analyze the complete diffusion path across all timesteps.

**Manifold-aware diffusion models** Multiple authors (Huang et al., 2022; De Bortoli et al., 2022; Thornton et al., 2022) have extended the theoretical framework of diffusion models to data supported on Riemannian manifolds. Our approach is different: we study the inherent geometric properties that emerge within the denoising landscape of standard diffusion models trained on data sampled from Euclidean spaces. Instead of enforcing Riemannian geometry, we use it as a tool to better understand conventional diffusion models.

**Understanding data manifold through score functions and geometry** Recent research explores how diffusion models relate to data geometry. Stanczuk et al. (2022), Kamkari et al. (2024), Ventura et al. (2025), and Humayun et al. (2024) studied how score functions capture manifold properties (through normal bundles, local intrinsic dimension, Jacobian spectra, and piecewise-linear approximations). Our work differs in two ways: we focus on information flow during denoising rather than data manifold structure, and we apply information geometry instead of analyzing score function spectra.

**Improving sampling strategy with geometry** Two recent works explore geometric formulations of diffusion models, aiming at improving sampling efficiency. Das et al. (2023) propose optimizing the forward noising process by following the shortest geodesic between  $p_0$  and  $p_t$  under the Fisher–Rao metric, hypothesizing that this minimizes accumulated



errors during denoising and thus indirectly improves sampling. In contrast, Ghimire et al. (2023) frame both the forward and reverse processes as Wasserstein gradient flows on probability spaces, using optimal transport geometry to propose a direct acceleration strategy. Our approach differs: we adopt an information geometric perspective (unlike Ghimire et al. (2023)) and focus on the reverse process (unlike Das et al. (2023)). Furthermore, Das et al. (2023) make a strong assumption that the data distribution  $p_0(x_0)$  is Gaussian, whereas we only assume that it has a density.

## 6 Limitations

Our geometric framework defines shortest paths (geodesics) between any two noisy samples, including those with very low noise levels (i.e., nearly clean data). However, by definition, the denoising distribution of a nearly clean sample approaches a Dirac delta, meaning it maps to essentially a single outcome. Since the Fisher-Rao metric locally approximates the KL divergence, and the KL divergence between two distinct Dirac delta distributions is infinite, distances between such low-noise samples become extremely large. As a result, optimizing geodesics between nearly clean samples becomes numerically unstable and impractical. Therefore, in our experiments, we select endpoints with a non-negligible level of noise to ensure tractable optimization (see Appendix E for details).

Another limitation comes from using Annealed Langevin Sampler (Algorithm 1) for transition path sampling, which works well in practice, but only approximately guarantees that points on the trajectory correctly follow  $p(\cdot|\gamma_s)$ . To ensure correct marginals with a single SDE, the change in the energy needs to be taken into account, as discussed by Albergo and Vanden-Eijnden (2024).

## 7 Conclusion

We proposed a novel perspective on the latent space of diffusion models by viewing it as a  $(D + 1)$ -dimensional statistical manifold, with the Fisher-Rao metric inducing a geometrical structure. By leveraging the fact that the denoising distributions form an exponential family, we showed that we can tractably estimate geodesics even for high-dimensional image diffusion models. We visualized our methods for image interpolations and demonstrated their utility in molecular transition path sampling.

This work deepens our understanding of the latent space in diffusion models and has the potential to inspire further research, including the development of novel applications of the spacetime geometric framework, such as enhanced sampling techniques.

## Acknowledgments

This work was supported by the Finnish Center for Artificial Intelligence (FCAI) under Flagship R5 (award 15011052). SH was supported by research grants from VILLUM FONDEN (42062), the Novo Nordisk Foundation through the Center for Basic Research in Life Science (NNF20OC0062606), and the European Research Council (ERC) under the European Union’s Horizon Programme (grant agreement 101125003). VG acknowledges the support from Saab-WASP (grant 411025), Academy of Finland (grant 342077), and the Jane and Aatos Erkko Foundation (grant 7001703).

## References

- Michael S Albergo and Eric Vanden-Eijnden. NETS: A non-equilibrium transport sampler. *arXiv*, 2024.
- Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.
- Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 1982.
- Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *ICLR*, 2018.
- Georgios Arvanitidis, Miguel González-Duque, Alison Pouplin, Dimitrios Kalatzis, and Soren Hauberg. Pulling back information geometry. In *AISTATS*, 2022.
- Nikolai Cencov. *Statistical decision rules and optimal inference*. American Mathematical Society, 1981.
- Ayan Das, Stathi Fotiadis, Anil Batra, Farhang Nabiei, FengTing Liao, Sattar Vakili, Da-shan Shiu, and Alberto Bernacchia. Image generation with shortest path diffusion. In *ICML*, 2023.

- Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *NeurIPS*, 2022.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Manfredo Perdigao Do Carmo and Flaherty Francis. *Riemannian geometry*. Springer, 1992.
- Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noe, Carla Gomes, Alan Aspuru-Guzik, and Kirill Neklyudov. Doob’s Lagrangian: A sample-efficient variational approach to transition path sampling. In *NeurIPS*, 2024.
- Bradley Efron. Tweedie’s formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614, 2011.
- Sandesh Ghimire, Jinyang Liu, Armand Comas, Davin Hill, Aria Masoomi, Octavia Camps, and Jennifer Dy. Geometry of score based generative models. *arXiv*, 2023.
- Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *ICLR*, 2019.
- Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary Markov processes. In *ICLR*, 2025.
- Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Bernd Ensing, and Max Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths. In *NeurIPS*, 2023.
- Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron Courville. Riemannian diffusion models. In *NeurIPS*, 2022.
- Ahmed Imtiaz Humayun, Ibtihel Amara, Cristina Nader Vasconcelos, Deepak Ramachandran, Candice Schumann, Junfeng He, Katherine A Heller, Golnoosh Farnadi, Negar Rostamzadeh, and Mohammad Havaei. What secrets do your manifolds hold? Understanding the local geometry of generative models. In *ICLR*, 2024.
- Michael Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics – Simulation and Computation*, 1989.
- Hamid Kamkari, Brendan Ross, Rasa Hosseinzadeh, Jesse Cresswell, and Gabriel Loaiza-Ganem. A geometric view of data complexity: Efficient local intrinsic dimension estimation with diffusion models. In *NeurIPS*, 2024.
- Rafał Karczewski, Markus Heinonen, and Vikas Garg. Devil is in the details: Density guidance for detail-aware generation with flow models. In *ICML*, 2025.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.
- Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024.
- Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *NeurIPS*, 2021.
- Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *NeurIPS*, 2023.
- Cheng Lu, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *ICML*, 2022.
- Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on Fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017.
- James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 2020.
- Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. In *NeurIPS*, 2021.

- Kumar Mishra, Ashok Kumar, and Ting-Kam Leonard Wong. Information geometry for the working information theorist. *arXiv*, 2023.
- Johannes Müller, Semih Çaycı, and Guido Montúfar. Fisher-Rao gradient flows of linear programs and state-action natural policy gradients. In *Symposium on Sparsity and Singular Structures*, 2024.
- Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 2020.
- Frank Nielsen and Vincent Garcia. Statistical exponential families: A digest with flash cards. *CORR*, 2009.
- OpenMP Architecture Review Board. OpenMP application program interface version 3.0, 2008. URL <http://www.openmp.org/mp-documents/spec30.pdf>.
- Yong-Hyun Park, Mingi Kwon, Jaewoong Choi, Junghyo Jo, and Youngjung Uh. Understanding the latent space of diffusion models through the lens of Riemannian geometry. In *NeurIPS*, 2023.
- Sanjeev Raja, Martin Sipka, Michael Psenka, Tobias Kreiman, Michal Pavelka, and Aditi Krishnapriyan. Action-minimization meets generative modeling: Efficient transition path sampling with the Onsager-Machlup functional. In *ICLR Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.
- Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs. In *ICLR*, 2025.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.
- Jan Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Your diffusion model secretly knows the dimension of the data manifold. *arXiv*, 2022.
- James Thornton, Michael Hutchinson, Emile Mathieu, Valentin De Bortoli, Yee Whye Teh, and Arnaud Doucet. Riemannian diffusion Schrödinger bridge. In *ICML Workshop Continuous Time Methods for Machine Learning*, 2022.
- Enrico Ventura, Beatrice Achilli, Gianluigi Silvestri, Carlo Lucibello, and Luca Ambrogioni. Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion. In *ICLR*, 2025.
- Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 2023.
- Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *ICML*, 2021.
- Qingtao Yu, Jaskirat Singh, Zhaoyuan Yang, Peter Henry Tu, Jing Zhang, Hongdong Li, Richard Hartley, and Dylan Campbell. Probability density geodesics in image diffusion latent space. In *CVPR*, 2025.

## A Notation

We denote  $\mathbf{x} \in \mathbb{R}^D$  a point in  $D$ -dimensional Euclidean space (a column vector),  $\text{Tr}(\mathbf{A}) = \sum_i A_{ii}$  - the trace operator of a square matrix  $\mathbf{A} \in \mathbb{R}^{k \times k}$ .

**Differential operators** For a scalar function  $f : \mathbb{R}^D \rightarrow \mathbb{R}, \mathbf{x} \mapsto f(\mathbf{x}) \in \mathbb{R}$ , we denote

$$\text{gradient: } \nabla_{\mathbf{x}} f(\tilde{\mathbf{x}}) = \left( \frac{\partial f}{\partial x^1}, \dots, \frac{\partial f}{\partial x^D} \right)^\top \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \in \mathbb{R}^D$$

$$\text{Hessian: } \nabla_{\mathbf{x}}^2 f(\tilde{\mathbf{x}}) = \left[ \frac{\partial^2 f}{\partial x^i \partial x^j} \right]_{i,j} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \in \mathbb{R}^{D \times D}$$

$$\text{Laplacian: } \Delta_{\mathbf{x}} f(\tilde{\mathbf{x}}) = \text{Tr}(\nabla_{\mathbf{x}}^2 f(\tilde{\mathbf{x}})) = \sum_{i=1}^D \frac{\partial^2 f}{\partial (x^i)^2} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \in \mathbb{R}.$$

For a curve  $\gamma : [0, 1] \rightarrow \mathbb{R}^k, s \mapsto \gamma_s \in \mathbb{R}^k$  we denote

$$\text{time derivative: } \dot{\gamma}_s = \frac{d}{ds} \gamma_s \in \mathbb{R}^k.$$

For a vector valued function  $f : \mathbb{R}^k \rightarrow \mathbb{R}^m, \mathbf{x} \mapsto (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))^\top \in \mathbb{R}^m$  we denote

$$\text{Jacobian: } \frac{\partial f(\tilde{\mathbf{x}})}{\partial \mathbf{x}} = \left[ \frac{\partial f^i}{\partial x^j} \right]_{i,j} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \in \mathbb{R}^{m \times k}$$

When  $k = m$ , we define

$$\text{divergence: } \text{div}_{\mathbf{x}} f(\tilde{\mathbf{x}}) = \text{Tr} \left( \frac{\partial f(\tilde{\mathbf{x}})}{\partial \mathbf{x}} \right) = \sum_{i=1}^k \frac{\partial f^i}{\partial x^i} \Big|_{\mathbf{x}=\tilde{\mathbf{x}}} \in \mathbb{R}$$

**Functions with two arguments** For  $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \rightarrow \mathbb{R}, (\mathbf{x}_1, \mathbf{x}_2) \mapsto f(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}$  we define (analogously w.r.t. second argument)

$$\text{gradient w.r.t. first argument: } \nabla_{\mathbf{x}_1} f(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = \left( \frac{\partial f}{\partial x_1^1}, \dots, \frac{\partial f}{\partial x_1^{k_1}} \right)^\top \Big|_{(\mathbf{x}_1, \mathbf{x}_2) = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2)} \in \mathbb{R}^{k_1}$$

For  $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \rightarrow \mathbb{R}^m, (\mathbf{x}_1, \mathbf{x}_2) \mapsto (f^1(\mathbf{x}_1, \mathbf{x}_2), \dots, f^m(\mathbf{x}_1, \mathbf{x}_2))^\top \in \mathbb{R}^m$  we define (analogously w.r.t. second argument)

$$\text{Jacobian w.r.t. first argument: } \frac{\partial f(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2)}{\partial \mathbf{x}_1} = \left[ \frac{\partial f^i}{\partial x_1^j} \right]_{i,j} \Big|_{(\mathbf{x}_1, \mathbf{x}_2) = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2)} \in \mathbb{R}^{m \times k_1}$$

## B Fisher-Rao, energy and KL functionals in exponential families

Throughout this section, we will work with the general form of an exponential family.

**Definition** (Exponential Family). *A parametric family of probability distributions  $\{p(\cdot|\boldsymbol{\theta})\}$  is called an exponential family if it can be expressed in the form*

$$p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\theta})^\top T(\mathbf{x}) - \psi(\boldsymbol{\theta})),$$

with  $\mathbf{x}$  a random variable modelling the data and  $\boldsymbol{\theta}$  the parameter of the distribution. In addition,  $T(\mathbf{x})$  is called a sufficient statistic,  $\boldsymbol{\eta}(\boldsymbol{\theta})$  natural (canonical) parameter,  $\psi(\boldsymbol{\theta})$  the log-partition (cumulant) function and  $h(\mathbf{x})$  is a base measure independent of  $\boldsymbol{\theta}$ .

### B.1 Fisher-Rao metric in exponential families

In this section, we will prove [Eq. 11](#), which is the Fisher-Rao metric applied to the case of exponential family.

**Proposition** (Fisher-Rao metric for an exponential family). *Let  $\{p(\cdot|\boldsymbol{\theta})\}$  be an exponential family. We denote  $\boldsymbol{\eta}(\boldsymbol{\theta})$  the natural parametrisation,  $T(\mathbf{x})$  the sufficient statistic and  $\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbb{E}[T(\mathbf{x})|\boldsymbol{\theta}]$  the expectation parameters. The Fisher-Rao metric is given by:*



$$\mathcal{I}(\boldsymbol{\theta}) = \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right),$$

where we equivalently write  $\mathcal{I}_{\boldsymbol{\theta}} = \mathcal{I}(\boldsymbol{\theta})$  to mean the metric at  $\boldsymbol{\theta}$ .

*Proof.* For  $p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\theta})^\top T(\mathbf{x}) - \psi(\boldsymbol{\theta}))$ , we have

$$\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \sum_k \eta^k(\boldsymbol{\theta}) T^k(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) = \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top T(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}). \quad (31)$$

Note that

$$\mathbb{E}[\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) | \boldsymbol{\theta}] = \int p(\mathbf{x}|\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = \int \nabla_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = \nabla_{\boldsymbol{\theta}} \int p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} = \mathbf{0}. \quad (32)$$

Therefore, by taking the expectation of both sides of Eq. 31, we get

$$\nabla_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) = \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top \boldsymbol{\mu}(\boldsymbol{\theta}), \quad (33)$$

where  $\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbb{E}[T(\mathbf{x})|\boldsymbol{\theta}]$ . Now we differentiate  $j$ -th component of both sides of Eq. 32 w.r.t  $\theta^i$ , and we get

$$\begin{aligned} 0 &= \frac{\partial}{\partial \theta^i} 0 = \frac{\partial}{\partial \theta^i} \mathbb{E} \left[ \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^j} | \boldsymbol{\theta} \right] = \frac{\partial}{\partial \theta^i} \int p(\mathbf{x}|\boldsymbol{\theta}) \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^j} d\mathbf{x} \\ &= \int \frac{\partial p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i} \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^j} d\mathbf{x} + \int p(\mathbf{x}|\boldsymbol{\theta}) \frac{\partial^2 \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} d\mathbf{x} \\ &= \mathbb{E} \left[ \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i} \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^j} | \boldsymbol{\theta} \right] + \mathbb{E} \left[ \frac{\partial^2 \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} | \boldsymbol{\theta} \right]. \end{aligned} \quad (34)$$

Therefore

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = \mathbb{E} \left[ \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i} \frac{\partial \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^j} | \boldsymbol{\theta} \right] = -\mathbb{E} \left[ \frac{\partial^2 \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} | \boldsymbol{\theta} \right]. \quad (35)$$

Now using Eq. 31, we have

$$\frac{\partial^2 \log p(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} = \frac{\partial}{\partial \theta^i} \left( \sum_k \frac{\partial \eta^k(\boldsymbol{\theta})}{\partial \theta^j} T^k(\mathbf{x}) - \frac{\partial \psi(\boldsymbol{\theta})}{\partial \theta^j} \right) = \sum_k \frac{\partial^2 \eta^k(\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} T^k(\mathbf{x}) - \frac{\partial^2 \psi(\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j}. \quad (36)$$

Therefore, from Eq. 35:

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = \frac{\partial^2 \psi(\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} - \sum_k \frac{\partial^2 \eta^k(\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} \mu^k(\boldsymbol{\theta}). \quad (37)$$

Now using Eq. 33, we have

$$\frac{\partial^2 \psi(\boldsymbol{\theta})}{\partial \theta^i \partial \theta^j} = \frac{\partial}{\partial \theta^j} \left( \sum_k \frac{\partial \eta^k(\boldsymbol{\theta})}{\partial \theta^i} \mu^k(\boldsymbol{\theta}) \right) = \sum_k \frac{\partial^2 \eta^k(\boldsymbol{\theta})}{\partial \theta^j \partial \theta^i} \mu^k(\boldsymbol{\theta}) + \sum_k \frac{\partial \eta^k(\boldsymbol{\theta})}{\partial \theta^i} \frac{\partial \mu^k(\boldsymbol{\theta})}{\partial \theta^j}. \quad (38)$$

Combining (Eq. 37) with (Eq. 38) yields:

$$\mathcal{I}_{ij}(\boldsymbol{\theta}) = \sum_k \frac{\partial \eta^k(\boldsymbol{\theta})}{\partial \theta^j} \frac{\partial \mu^k(\boldsymbol{\theta})}{\partial \theta^i} = \left[ \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) \right]_{ij}. \quad (39)$$

□

## B.2 Energy function in exponential families

**Proposition** (Energy function for an exponential family). *Let  $\gamma : [0, 1] \rightarrow \Theta$  be a smooth curve in the parameter space  $\Theta$  of an exponential family, and let  $\mathcal{I}(\theta)$  be the Fisher-Rao metric on  $\Theta$ . Then*

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_0^1 \left( \frac{d}{ds} \boldsymbol{\eta}(\gamma_s) \right)^\top \left( \frac{d}{ds} \boldsymbol{\mu}(\gamma_s) \right) ds.$$

*Proof.* We know that the energy of  $\gamma$  can be defined as  $\mathcal{E}(\gamma) = \frac{1}{2} \int_0^1 \|\dot{\gamma}_s\|_{\mathcal{I}}^2 ds$ . We replace the Riemannian metric  $\mathcal{I}$  with the previously obtained expression of the Fisher-Rao metric (Eq. 11).

$$\begin{aligned} \mathcal{E}(\gamma) &= \frac{1}{2} \int_0^1 \|\dot{\gamma}_s\|_{\mathcal{I}}^2 ds = \frac{1}{2} \int_0^1 \dot{\gamma}_s^\top \mathcal{I}_{\gamma_s} \dot{\gamma}_s ds \\ &= \frac{1}{2} \int_0^1 \dot{\gamma}(s)^\top \left( \frac{\partial \boldsymbol{\eta}(\gamma_s)}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\gamma_s)}{\partial \boldsymbol{\theta}} \right) \dot{\gamma}_s ds \\ &= \frac{1}{2} \int_0^1 \left( \frac{\partial \boldsymbol{\eta}(\gamma_s)}{\partial \boldsymbol{\theta}} \dot{\gamma}_s \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\gamma_s)}{\partial \boldsymbol{\theta}} \dot{\gamma}_s \right) ds \\ &= \frac{1}{2} \int_0^1 \left( \frac{d}{ds} \boldsymbol{\eta}(\gamma_s) \right)^\top \left( \frac{d}{ds} \boldsymbol{\mu}(\gamma_s) \right) ds. \end{aligned}$$

□

## B.3 Kullback-Leibler divergence in exponential families

As in Eq. 6, the Fisher-Rao metric is the local approximation of the KL divergence, i.e.

$$\text{KL}(p(\cdot|\boldsymbol{\theta}_1)||p(\cdot|\boldsymbol{\theta}_2)) \approx \frac{1}{2} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \mathcal{I}_{\boldsymbol{\theta}_1} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2).$$

In the case of exponential families, we have  $\mathcal{I}_{\boldsymbol{\theta}} = \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)$ , and thus we can write

$$\begin{aligned} \text{KL}(p(\cdot|\boldsymbol{\theta}_1)||p(\cdot|\boldsymbol{\theta}_2)) &\approx \frac{1}{2} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}} \right) (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \\ &\approx \frac{1}{2} (\boldsymbol{\eta}(\boldsymbol{\theta}_1) - \boldsymbol{\eta}(\boldsymbol{\theta}_2))^\top (\boldsymbol{\mu}(\boldsymbol{\theta}_1) - \boldsymbol{\mu}(\boldsymbol{\theta}_2)). \end{aligned}$$

It turns out that the RHS always corresponds to a notion of distribution divergence (not only when  $\boldsymbol{\theta}_1$  and  $\boldsymbol{\theta}_2$  are close together), namely the *symmetrized* Kullback-Leibler divergence:

$$\text{KL}^S(p||q) := \frac{1}{2} (\text{KL}(p||q) + \text{KL}(q||p)). \quad (40)$$

**Lemma 1** (KL in exponential families). *Let  $\mathcal{P} = \{p(\cdot|\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$  be an exponential family with  $p(\mathbf{x}|\boldsymbol{\theta}) = h(\mathbf{x}) \exp(\boldsymbol{\eta}(\boldsymbol{\theta})^\top T(\mathbf{x}) - \psi(\boldsymbol{\theta}))$ , and  $\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})}[T(\mathbf{x})]$ . Then*

$$\text{KL}(\boldsymbol{\theta}_1||\boldsymbol{\theta}_2) = (\boldsymbol{\eta}(\boldsymbol{\theta}_1) - \boldsymbol{\eta}(\boldsymbol{\theta}_2))^\top \boldsymbol{\mu}(\boldsymbol{\theta}_1) - \psi(\boldsymbol{\theta}_1) + \psi(\boldsymbol{\theta}_2), \quad (41)$$

where we abuse notation and write  $\text{KL}(\boldsymbol{\theta}_1||\boldsymbol{\theta}_2)$  instead of  $\text{KL}(p(\cdot|\boldsymbol{\theta}_1)||p(\cdot|\boldsymbol{\theta}_2))$ .

*Proof.*

$$\begin{aligned} \text{KL}(\boldsymbol{\theta}_1||\boldsymbol{\theta}_2) &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}_1)} [\log p(\mathbf{x}|\boldsymbol{\theta}_1) - \log p(\mathbf{x}|\boldsymbol{\theta}_2)] \\ &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}_1)} [\boldsymbol{\eta}(\boldsymbol{\theta}_1)^\top T(\mathbf{x}) - \boldsymbol{\eta}(\boldsymbol{\theta}_2)^\top T(\mathbf{x}) - \psi(\boldsymbol{\theta}_1) + \psi(\boldsymbol{\theta}_2)] \\ &= (\boldsymbol{\eta}(\boldsymbol{\theta}_1) - \boldsymbol{\eta}(\boldsymbol{\theta}_2))^\top \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}_1)} [T(\mathbf{x})] - \psi(\boldsymbol{\theta}_1) + \psi(\boldsymbol{\theta}_2) \\ &= (\boldsymbol{\eta}(\boldsymbol{\theta}_1) - \boldsymbol{\eta}(\boldsymbol{\theta}_2))^\top \boldsymbol{\mu}(\boldsymbol{\theta}_1) - \psi(\boldsymbol{\theta}_1) + \psi(\boldsymbol{\theta}_2). \end{aligned}$$

□

**Lemma 2** (Symmetrized KL in exponential families). *With assumptions of Lemma 1, we have*

$$\text{KL}^S(\theta_1 || \theta_2) = \frac{1}{2} (\eta(\theta_1) - \eta(\theta_2))^\top (\mu(\theta_1) - \mu(\theta_2)). \quad (42)$$

*Proof.*

$$\begin{aligned} 2 \text{KL}^S(\theta_1 || \theta_2) &= \text{KL}(\theta_1 || \theta_2) + \text{KL}(\theta_2 || \theta_1) \\ &= (\eta(\theta_1) - \eta(\theta_2))^\top \mu(\theta_1) - \cancel{\psi(\theta_1)} + \cancel{\psi(\theta_2)} + (\eta(\theta_2) - \eta(\theta_1))^\top \mu(\theta_2) - \cancel{\psi(\theta_2)} + \cancel{\psi(\theta_1)} \\ &= (\eta(\theta_1) - \eta(\theta_2))^\top (\mu(\theta_1) - \mu(\theta_2)). \end{aligned}$$

□

The formula for KL in Lemma 1 is not useful in practice, because it requires knowing  $\psi(\theta)$ , which can be unknown or expensive to evaluate. However, the gradients with respect to both arguments depend only on  $\eta$  and  $\mu$ .

**Lemma 3** (KL gradients). *With assumptions of Lemma 1, we have for any  $\theta_1, \theta_2$*

$$\begin{aligned} \nabla_{\theta_1} \text{KL}(\theta_1 || \theta_2) &= \frac{\partial \mu(\theta_1)}{\partial \theta}^\top (\eta(\theta_1) - \eta(\theta_2)) \\ \nabla_{\theta_2} \text{KL}(\theta_1 || \theta_2) &= \frac{\partial \eta(\theta_2)}{\partial \theta}^\top (\mu(\theta_2) - \mu(\theta_1)) \end{aligned} \quad (43)$$

*Proof.* The proof is a straightforward calculation using Lemma 1 and Eq. 33. We have

$$\begin{aligned} \nabla_{\theta_1} \text{KL}(\theta_1 || \theta_2) &= \nabla_{\theta_1} \left( (\eta(\theta_1) - \eta(\theta_2))^\top \mu(\theta_1) - \psi(\theta_1) + \psi(\theta_2) \right) \\ &= \frac{\partial \eta(\theta_1)}{\partial \theta}^\top \mu(\theta_1) + \frac{\partial \mu(\theta_1)}{\partial \theta}^\top (\eta(\theta_1) - \eta(\theta_2)) - \nabla_{\theta} \psi(\theta_1) \\ &\stackrel{(33)}{=} \cancel{\frac{\partial \eta(\theta_1)}{\partial \theta}^\top \mu(\theta_1)} + \frac{\partial \mu(\theta_1)}{\partial \theta}^\top (\eta(\theta_1) - \eta(\theta_2)) - \cancel{\frac{\partial \eta(\theta_1)}{\partial \theta}^\top \mu(\theta_1)} \\ &= \frac{\partial \mu(\theta_1)}{\partial \theta}^\top (\eta(\theta_1) - \eta(\theta_2)) \end{aligned}$$

and

$$\begin{aligned} \nabla_{\theta_2} \text{KL}(\theta_1 || \theta_2) &= \nabla_{\theta_2} \left( (\eta(\theta_1) - \eta(\theta_2))^\top \mu(\theta_1) - \psi(\theta_1) + \psi(\theta_2) \right) \\ &\stackrel{(33)}{=} -\frac{\partial \eta(\theta_2)}{\partial \theta}^\top \mu(\theta_1) + \frac{\partial \eta(\theta_2)}{\partial \theta}^\top \mu(\theta_2) = \frac{\partial \eta(\theta_2)}{\partial \theta}^\top (\mu(\theta_2) - \mu(\theta_1)) \end{aligned}$$

□

Knowing the gradients allows for estimating the KL divergence along a curve.

**Proposition 2** (KL along a curve). *Let  $\gamma : [0, 1] \rightarrow \Theta$  be a smooth denoising curve, and  $\theta^* \in \Theta$ . Then:*

$$\begin{aligned} \text{KL}(\gamma_s || \theta^*) &= \text{KL}(\gamma_0 || \theta^*) + \int_0^s \left( \frac{d}{du} \mu(\gamma_u) \right)^\top (\eta(\gamma_u) - \eta(\theta^*)) du \\ \text{KL}(\theta^* || \gamma_s) &= \text{KL}(\theta^* || \gamma_0) + \int_0^s \left( \frac{d}{du} \eta(\gamma_u) \right)^\top (\mu(\gamma_u) - \mu(\theta^*)) du \end{aligned} \quad (44)$$

*Proof.*

$$\begin{aligned}
& \text{KL}(\gamma_s || \theta^*) - \text{KL}(\gamma_0 || \theta^*) = \\
&= \int_0^s \frac{d}{du} (\text{KL}(\gamma_u || \theta^*)) du && // \text{Fundamental theorem of calculus} \\
&= \int_0^s \nabla_{\theta^*} \text{KL}(\gamma_u || \theta^*)^\top \dot{\gamma}_u du && // \text{Chain rule} \\
&= \int_0^s \left( \frac{\partial \mu(\gamma_u)}{\partial \theta} \dot{\gamma}_u \right)^\top (\eta(\gamma_u) - \eta(\theta^*)) du && // \text{Lemma 3} \\
&= \int_0^s \left( \frac{d}{du} \mu(\gamma_u) \right)^\top (\eta(\gamma_u) - \eta(\theta^*)) du && // \text{Chain rule.}
\end{aligned}$$

Using the same reasoning we have

$$\begin{aligned}
& \text{KL}(\theta^* || \gamma_s) - \text{KL}(\theta^* || \gamma_0) = \int_0^s \frac{d}{du} (\text{KL}(\theta^* || \gamma_u)) du \\
&= \int_0^s \nabla_{\theta^*} \text{KL}(\theta^* || \gamma_u)^\top \dot{\gamma}_u du \\
&= \int_0^s \left( \frac{\partial \eta(\gamma_u)}{\partial \theta} \dot{\gamma}_u \right)^\top (\mu(\gamma_u) - \mu(\theta^*)) du \\
&= \int_0^s \left( \frac{d}{du} \eta(\gamma_u) \right)^\top (\mu(\gamma_u) - \mu(\theta^*)) du
\end{aligned}$$

□

## C Computational complexity of ODE pullback metrics

In this section, we elaborate on why the information geometric approach is computationally significantly less demanding than the pullback metric approach.

**Energy estimation** Given a latent generative model with a decoder  $f : \mathbb{R}^k \rightarrow \mathbb{R}^D, z \mapsto f(z)$ , the pullback metric is defined as (Arvanitidis et al., 2018)

$$M_z = \frac{\partial f(z)}{\partial z}^\top \frac{\partial f(z)}{\partial z} \in \mathbb{R}^{k \times k} \quad (45)$$

and the Riemannian norm of a tangent vector  $v$ :

$$\|v\|_{M_z}^2 = v^\top M_z v = \left\| \frac{\partial f(z)}{\partial z} v \right\|^2, \quad (46)$$

where the last norm is Euclidean. The Riemannian energy of a latent curve  $\gamma : [0, 1] \rightarrow \mathbb{R}^k$  is given by

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_0^1 \|\dot{\gamma}_s\|_{M_{\gamma_s}}^2 ds = \frac{1}{2} \int_0^1 \left\| \frac{\partial f(\gamma_s)}{\partial z} \dot{\gamma}_s \right\|^2 ds = \frac{1}{2} \int_0^1 \left\| \frac{d}{ds} f(\gamma_s) \right\|^2 ds, \quad (47)$$

where the last equality follows from the chain rule. Therefore, in practice the energy of a latent curve is approximated with finite differences of the discretized curve:  $\gamma_n := \gamma_{s_n}$  for  $s_n = \frac{n}{N-1}, n = 0, \dots, N-1$ , and  $ds = \frac{1}{N-1}$ , we have:

$$\mathcal{E}(\gamma) \approx \frac{1}{2ds} \sum_{n=0}^{N-2} \|f(\gamma_{n+1}) - f(\gamma_n)\|^2. \quad (48)$$

Therefore, estimation of the energy of a latent curve requires  $N$  evaluations of the decoder  $f$  when the curve is discretized into  $N$  points.



**Spacetime: pullback vs information geometry** In the case of the diffusion’s latent spacetime, the decoder  $f(\mathbf{x}_t) = \mathbf{x}_0^{\text{PF}}(\mathbf{x}_t, t)$  is the solution of the PF-ODE (Eq. 4) solved from  $t$  to 0. Therefore, estimating the energy in the pullback geometry requires solving the PF-ODE  $N$  times. And each solver step requires evaluating the score function (or equivalently, the denoiser model  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ ).

On the other hand, as we show in Equation (Eq. 23), estimating the latent energy using information geometry requires  $N$  estimates of the expectation parameter  $\mu$ :

$$\mu(\mathbf{x}_t, t) \approx \left( \hat{\mathbf{x}}_0(\mathbf{x}_t), \frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \hat{\mathbf{x}}_0(\mathbf{x}_t) + \|\hat{\mathbf{x}}_0(\mathbf{x}_t)\|^2 \right), \quad (49)$$

which can be efficiently estimated with a single Jacobian-Vector Product (JVP), as we show in Listing 1. Therefore, the total cost of estimating the energy of a curve discretized into  $N$  points is:

$$\begin{aligned} \text{Pullback geometry: } & \mathcal{O}(NK) \text{ evaluations of } \hat{\mathbf{x}}_0(\mathbf{x}_t) \\ \text{Information geometry: } & \mathcal{O}(N) \text{ JVPs of } \hat{\mathbf{x}}_0(\mathbf{x}_t) \end{aligned}$$

where  $K$  is the number of solver steps. Given that the JVP costs roughly twice as much as evaluating the denoiser (Meng et al., 2021), and  $K \gg 2$ , the information geometry approach is significantly more efficient. This is especially important since we usually need to not only evaluate the energy, but also optimize it (differentiate it many times) to find geodesics.

## D Probabilistic structure of the denoising process

In this section, we show that denoising distributions form an exponential family, and we identify closed-form expressions of the key parameters. The main theorem is established as a concatenation of two lemmas. In Lemma 4, we derive the sufficient statistic and natural parameters; and in Lemma 5, we compute the expectation parameter using the first and second denoising moments.

**Proposition 1** (Exponential family of denoising). *Let  $\mathbf{x}_t$  be a noisy observation corresponding to diffusion time  $t$ , as introduced in Eq. 1. Then*

$$p(\mathbf{x}_0 | \mathbf{x}_t) = h(\mathbf{x}_0) \exp \left( \boldsymbol{\eta}(\mathbf{x}_t, t)^\top T(\mathbf{x}_0) - \psi(\mathbf{x}_t, t) \right), \quad (17)$$

with  $h = q$  the data distribution density,  $\psi$  the log-partition function, and

$$\boldsymbol{\eta}(\mathbf{x}_t, t) = \left( \frac{\alpha_t}{\sigma_t^2} \mathbf{x}_t, -\frac{\alpha_t^2}{2\sigma_t^2} \right) \quad (\text{natural parameter}) \quad (18)$$

$$T(\mathbf{x}_0) = (\mathbf{x}_0, \|\mathbf{x}_0\|^2) \quad (\text{sufficient statistic}) \quad (19)$$

$$\mu(\mathbf{x}_t, t) = \underbrace{\left( \frac{1}{\alpha_t} \left( \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \right) \right)}_{\text{'space': } \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]} \underbrace{\left( \frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 \right)}_{\text{'time': } \mathbb{E}[\|\mathbf{x}_0\|^2 | \mathbf{x}_t]} \quad (20)$$

*Proof.* See Lemma 4 for the derivation of the natural parameter  $\boldsymbol{\eta}$  and the sufficient statistic  $T$ , and Lemma 5, for the derivation of the expectation parameter  $\mu$ .  $\square$

### D.1 Denoising distributions as an exponential family

**Lemma 4** (Denoising distribution as exponential families). *Let  $\mathbf{x}_t$  be a noised observation of a latent variable  $\mathbf{x}_0$  under a known diffusion process at time  $t$ . Then the denoising distributions  $p(\mathbf{x}_0 | \mathbf{x}_t)$  form an exponential family*

$$p(\mathbf{x}_0 | \mathbf{x}_t) = h(\mathbf{x}_0) \exp \left( \boldsymbol{\eta}(\mathbf{x}_t, t)^\top T(\mathbf{x}_0) - \psi(\mathbf{x}_t, t) \right), \quad (50)$$

with  $h$  the base measure and  $\psi$  the log-partition function.

The sufficient statistics  $T$  and the natural parameters  $\boldsymbol{\eta}$  are given by

$$T(\mathbf{x}_0) = (\mathbf{x}_0, \|\mathbf{x}_0\|^2) \quad \text{and} \quad \boldsymbol{\eta}(\mathbf{x}_t, t) = \left( \frac{\alpha_t}{\sigma_t^2} \mathbf{x}_t, -\frac{\alpha_t^2}{2\sigma_t^2} \right). \quad (51)$$

*Proof.* The denoising distribution is given by

$$p(\mathbf{x}_0 | \mathbf{x}_t) = \frac{p(\mathbf{x}_t | \mathbf{x}_0) q(\mathbf{x}_0)}{p_t(\mathbf{x}_t)},$$

where  $q$  is the data distribution,  $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$  is the forward density (Eq. 1), and  $p_t(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_0)q(\mathbf{x}_0)d\mathbf{x}_0$  is the marginal distribution at time  $t$ . Therefore

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{x}_0) &= \frac{1}{(2\pi\sigma_t^2)^{D/2}} \exp\left(-\frac{\|\mathbf{x}_t - \alpha_t\mathbf{x}_0\|^2}{2\sigma_t^2}\right) \\ &= \frac{1}{(2\pi\sigma_t^2)^{D/2}} \exp\left(-\frac{\|\mathbf{x}_t\|^2}{2\sigma_t^2} + \frac{\alpha_t}{\sigma_t^2}\mathbf{x}_t^\top\mathbf{x}_0 - \frac{\alpha_t^2}{2\sigma_t^2}\|\mathbf{x}_0\|^2\right) \\ &= \exp\left(-\frac{D}{2}\log(2\pi\sigma_t^2) - \frac{\|\mathbf{x}_t\|^2}{2\sigma_t^2}\right) \exp\left(-\frac{\alpha_t^2}{2\sigma_t^2}\|\mathbf{x}_0\|^2 + \frac{\alpha_t}{\sigma_t^2}\mathbf{x}_t^\top\mathbf{x}_0\right). \end{aligned} \quad (52)$$

By substituting into the denoising density, we get

$$\begin{aligned} p(\mathbf{x}_0|\mathbf{x}_t) &= q(\mathbf{x}_0) \exp\left\{-\frac{\alpha_t^2}{2\sigma_t^2}\|\mathbf{x}_0\|^2 + \frac{\alpha_t}{\sigma_t^2}\mathbf{x}_t^\top\mathbf{x}_0 - \left(\log p_t(\mathbf{x}_t) + \frac{D}{2}\log(2\pi\sigma_t^2) + \frac{\|\mathbf{x}_t\|^2}{2\sigma_t^2}\right)\right\} \\ &= h(\mathbf{x}_0) \exp\left(\boldsymbol{\eta}(\mathbf{x}_t)^\top T(\mathbf{x}_0) - \psi(\mathbf{x}_t)\right), \end{aligned} \quad (53)$$

where

$$\boldsymbol{\eta}(\mathbf{x}_t, t) = \left(\frac{\alpha_t}{\sigma_t^2}\mathbf{x}_t, -\frac{\alpha_t^2}{2\sigma_t^2}\right) \in \mathbb{R}^{D+1} \quad (54)$$

$$T(\mathbf{x}_0) = (\mathbf{x}_0, \|\mathbf{x}_0\|^2) \in \mathbb{R}^{D+1} \quad (55)$$

$$h(\mathbf{x}_0) = q(\mathbf{x}_0) \in \mathbb{R} \quad (56)$$

$$\psi(\mathbf{x}_t, t) = \log p_t(\mathbf{x}_t) + \frac{D}{2}\log(2\pi\sigma_t^2) + \frac{\|\mathbf{x}_t\|^2}{2\sigma_t^2} \in \mathbb{R} \quad (57)$$

□

Note that, if the data distribution is Boltzmann, i.e.  $q(\mathbf{x}_0) \propto \exp(-U(\mathbf{x}_0))$  for some energy function  $U$ , we have:

$$\begin{aligned} p(\mathbf{x}_0|\mathbf{x}_t) &\propto q(\mathbf{x}_0)p(\mathbf{x}_t|\mathbf{x}_0) \propto \exp(-(U(\mathbf{x}_0))) \exp\left(-\frac{\|\mathbf{x}_t - \alpha_t\mathbf{x}_0\|^2}{2\sigma_t^2}\right) \\ &= \exp\left(-U(\mathbf{x}_0) - \frac{1}{2}\text{SNR}(t)\|\mathbf{x}_0 - \mathbf{x}_t/\alpha_t\|^2\right). \end{aligned}$$

This implies that  $p(\mathbf{x}_0|\mathbf{x}_t)$  is also a Boltzmann distribution with  $p(\mathbf{x}_0|\mathbf{x}_t) \propto \exp(-U(\mathbf{x}_0|\mathbf{x}_t))$  for

$$U(\mathbf{x}_0|\mathbf{x}_t) = U(\mathbf{x}_0) + \frac{1}{2}\text{SNR}(t)\|\mathbf{x}_0 - \mathbf{x}_t/\alpha_t\|^2. \quad (58)$$

## D.2 Second denoising moment derivation

We can now derive the expectation parameter  $\boldsymbol{\mu}(\mathbf{x}_t, t)$ , which is required to compute the geodesic energy (see Eq. 13). The following proposition provides a closed-form expression for  $\boldsymbol{\mu}$  in terms of the denoising moments:

**Lemma 5** (Expectation parameter of a denoising distribution). *proposition Let  $\boldsymbol{\mu}(\mathbf{x}_t, t) := \mathbb{E}[T(\mathbf{x}_0) | \mathbf{x}_t]$  denote the expectation parameter corresponding to the sufficient statistics  $T$ . Then,*

$$\boldsymbol{\mu}(\mathbf{x}_t, t) = \left(\frac{1}{\alpha_t}(\mathbf{x}_t + \sigma_t^2\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)), \frac{\sigma_t^2}{\alpha_t}\text{div}_{\mathbf{x}_t}\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2\right), \quad (59)$$

where the first component corresponds to the spatial parameter  $\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]$ , and the second to the temporal parameter  $\mathbb{E}[\|\mathbf{x}_0\|^2 | \mathbf{x}_t]$ .

*Proof.* From Lemma 4, we already have the expression of the sufficient statistic  $T(\mathbf{x}_0) = (\mathbf{x}_0, \|\mathbf{x}_0\|^2)$ . We need to derive the first  $\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]$  and second  $\mathbb{E}[\|\mathbf{x}_0\|^2 | \mathbf{x}_t]$  denoising moments.

Recall that the forward corruption process is  $p(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$ , the score of the marginal can be expressed using Tweedie's formula (Efron, 2011):

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = -\frac{1}{\sigma_t^2}(\mathbf{x}_t - \alpha_t\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]) \quad (60)$$

It follows directly that the first denoising moment is:

$$\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] = \frac{1}{\alpha_t} (\mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)). \quad (61)$$

The denoising covariance is known (Meng et al., 2021):

$$\text{Cov}[\mathbf{x}_0 | \mathbf{x}_t] = \frac{\sigma_t^2}{\alpha_t^2} (\mathbf{I} + \sigma_t^2 \nabla_{\mathbf{x}_t}^2 \log p_t(\mathbf{x}_t)). \quad (62)$$

Therefore, from the definition of conditional variance, we can deduce the second denoising moment:

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}_0\|^2 | \mathbf{x}_t] &= \mathbb{E}[\|\mathbf{x}_0 - \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 | \mathbf{x}_t] + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 \\ &= \text{Tr}(\text{Cov}[\mathbf{x}_0 | \mathbf{x}_t]) + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 \\ &\stackrel{(62)}{=} \frac{\sigma_t^2}{\alpha_t^2} (D + \sigma_t^2 \Delta \log p_t(\mathbf{x}_t)) + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 \\ &= \frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \left( \frac{\mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}{\alpha_t} \right) + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2 \\ &= \frac{\sigma_t^2}{\alpha_t} \text{div}_{\mathbf{x}_t} \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t] + \|\mathbb{E}[\mathbf{x}_0 | \mathbf{x}_t]\|^2. \end{aligned} \quad (63)$$

□

## E Experimental details

### E.1 Toy Gaussian mixture

For the experiments with a 1D Gaussian mixture (Fig. 1, and Fig. 5 left), we define the data distribution as  $p_0 = \sum_{i=1}^3 \pi_i \mathcal{N}(\mu_i, \sigma^2)$  with  $\mu_1 = -2.5, \mu_2 = 0.5, \mu_3 = 2.5, \pi_1 = 0.275, \pi_2 = 0.45, \pi_3 = 0.275$ , and  $\sigma = 0.75$ . We specify the forward process (Eq. 1) as Variance-Preserving (Song et al., 2021b), i.e. satisfying  $\alpha_t^2 + \sigma_t^2 = 1$ , and assume as log-SNR linear noise schedule, i.e.  $\lambda_t = \log \text{SNR}(t) = \lambda_{\max} + (\lambda_{\min} - \lambda_{\max})t$  for  $\lambda_{\min} = -10, \lambda_{\max} = 10$ . Which implies:  $\alpha_t^2 = \text{sigmoid}(\lambda_t), \sigma_t^2 = \text{sigmoid}(-\lambda_t)$ .

Since  $p_0$  is a Gaussian mixture, all marginals  $p_t$  are also Gaussian mixtures, and training a diffusion model is unnecessary, as the score function  $\nabla_{\mathbf{x}} \log_t(\mathbf{x})$  is known analytically. In this example, the data is 1D, and the spacetime is 2D.

To generate Fig. 1 we estimate the geodesic between  $\theta_1 = (-2.3, 0.35)$ , and  $\theta_2 = (2, 0.4)$  by parametrizing  $\gamma$  with a cubic spline (Arvanitidis et al., 2022) with two nodes, and discretizing it into  $N = 128$  points and taking 1000 optimization steps with Adam optimizer and learning rate  $\eta = 0.1$ , which takes a few seconds on an M1 CPU.

To generate Fig. 5 left, we generate 3 PF-ODE sampling trajectories starting from  $x = 1, 0, -1$  using an Euler solver with 512 solver steps. We solve only until  $t = t_{\min} = 0.1$  (as opposed to  $t = 0$ ), because for  $t \approx 0$ , the denoising distributions  $p(\mathbf{x}_0 | \mathbf{x}_t)$  become closer to Dirac delta distributions  $\delta_{\mathbf{x}_t}$ , which makes the energies very large. For each sampling trajectory, we take the endpoints  $(x_1, 1), (x_{t_{\min}}, t_{\min})$  and estimate the geodesic between them using Eq. 23 with a cubic spline with 10 nodes, discretizing it into 512 points, and taking 2000 gradient steps of AdamW optimizer with learning rate  $\eta = 0.01$ . This takes roughly 10 seconds on an M1 CPU.

### E.2 Image data

For all experiments on image data, we use the pretrained EDM2 model trained on ImageNet512 (Karras et al., 2024) (specifically, the edm2-img512-xxl-fid checkpoint), which is a Variance-Exploding model, i.e.  $\alpha_t = 1$ , and using the noise schedule  $\sigma_t = t$ . It is a latent diffusion model, using a fixed StabilityVAE (Rombach et al., 2022) as the encoder/decoder.

**Image interpolations** To interpolate between to images, we encode them with StabilityVAE to obtain two latent codes  $\mathbf{x}_0^1, \mathbf{x}_0^2$ , and encode them both with PF-ODE (Eq. 4) from  $t = 0$  to  $t = t_{\min} = 0.368$ , corresponding to  $\log \text{SNR}(t_{\min}) = 2$ . This is to avoid very high values of energy for  $t \approx 0$ . We then optimize the geodesic between  $(\mathbf{x}_{t_{\min}}^1, t_{\min})$  and  $(\mathbf{x}_{t_{\min}}^2, t_{\min})$  by parametrizing it with a cubic spline with 8 nodes, and minimizing Eq. 23 using AdamW optimizer with learning rate  $\eta = 0.1$  in two stages:

- Coarse optimization: discretizing the curve into 16 points, and taking 350 gradient steps;
- Finetuning: discretizing the curve into 64 steps, and taking 250 gradient steps.

This procedure takes roughly 50 minutes on an A100 NVIDIA GPU per interpolation image pair.

Each interpolating geodesic  $\gamma$  has two components:  $\gamma_s = (\mathbf{x}_s, t(s))$ . To produce Fig. 4, we chose  $s_{\max} = \arg \max_s t(s)$  and visualize  $\mathbf{x}_{s_{\max}}$ , and  $\sigma_{\max} = t(s_{\max})$ .

**PF-ODE sampling trajectories** To generate PF-ODE sampling trajectories, we use the 2nd order Heun solver (Karras et al., 2022) with 64 steps, and solve from  $t = 80$  to  $t_{\min} = 0.135$  corresponding to  $\log \text{SNR}(t_{\min}) = 4$ . This is to avoid instabilities for small  $t$ . We parametrize the geodesic directly with the entire sampling trajectory  $\gamma_t = (\mathbf{x}_t, t)$  for  $t = T, \dots, t_{\min}$ , where the  $t$  schedule corresponds to EDM2 model’s sampling schedule.

We then fix the endpoints of the trajectory, and optimize the intermediate points using AdamW optimizer with learning rate  $\eta = 0.0001$  (larger learning rates lead to NaN values) and take 600 optimization steps. This procedure took roughly 2 hours on an A100 NVIDIA GPU per a single sampling trajectory.

To visualize intermediate noisy images at diffusion time  $t$ , we rescale them with  $\frac{\sigma_{\text{data}}}{\sqrt{\sigma_{\text{data}}^2 + \sigma_t^2}}$  before decoding with the VAE decoder, to avoid unrealistic color values, where we set  $\sigma_{\text{data}} = 0.5$  as in Karras et al. (2022).

### E.3 Molecular data

**Approximating the base energy function with a neural network** We follow Holdijk et al. (2023) and represent the energy function of Alanine Dipeptide in the space of two dihedral angles  $\phi, \psi \in [-\pi, \pi]$ . We use the code provided by the authors at [github.com/LarsHoldijk/SOCTransitionPaths](https://github.com/LarsHoldijk/SOCTransitionPaths), which estimates the energy  $U(\phi, \psi)$ . However, even though the values of the energy  $U$  looked reasonably, we found that the provided implementation of  $\frac{\partial U}{\partial \phi}$ , and  $\frac{\partial U}{\partial \psi}$  yielded unstable results due to discontinuities.

Instead, we trained an auxiliary feedforward neural network  $U_\theta$  to approximate  $U$ . We parametrized with two hidden layers of size 64 with SiLU activation functions, and trained it on a uniformly discretized grid  $[-\pi, \pi] \times [-\pi, \pi]$  into 16384 points. We trained the model with mean squared error for 8192 steps using Adam optimizer with a learning rate  $\eta = 0.001$  until the model converged to an average loss of  $\approx 1.5$ . This took approximately two and a half minutes on an M1 CPU. In the subsequent experiments, we estimate  $\nabla_{\mathbf{x}} U(\mathbf{x})$  with automatic differentiation on the trained auxiliary model.

**Generating samples from the energy landscape** To generate samples from the data distribution  $p_0(\mathbf{x}_0) \propto \exp(-U(\mathbf{x}_0))$ , we initialize the samples uniformly on the  $[-\pi, \pi] \times [-\pi, \pi]$  grid, and use Langevin dynamics

$$d\mathbf{x} = -\nabla_{\mathbf{x}} U(\mathbf{x})dt + \sqrt{2}dW_t \quad (64)$$

with the Euler-Maruyama solver for  $dt = 0.001$  and  $N = 1000$  steps.

**Training a diffusion model on the energy landscape** To estimate the spacetime geodesics, we need a denoiser network approximating the denoising mean  $\hat{\mathbf{x}}_0(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ . We parametrize the denoiser network with

```
from ddpm import MLP
model = MLP(
    hidden_size=128,
    hidden_layers=3,
    emb_size=128,
    time_emb="sinusoidal",
    input_emb="sinusoidal"
)
```

using the TinyDiffusion implementation [github.com/tanelp/tiny-diffusion](https://github.com/tanelp/tiny-diffusion). We trained the model using the weighted denoising loss:  $w(\lambda_t)\|\hat{\mathbf{x}}_0(\mathbf{x}_t, t) - \mathbf{x}_0\|^2$  with a weight function  $w(\lambda_t) = \sqrt{\text{sigmoid}(\lambda_t + 2)}$  and an adaptive noise schedule (Kingma and Gao, 2023). We train the model for 4000 steps using the AdamW optimizer with learning rate  $\eta = 0.001$ , which took roughly 1 minute on an M1 CPU.

**Spacetime geodesics** With a trained denoiser  $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ , we can estimate the expectation parameter  $\mu$  (Eq. 22) and thus curves energies in the spacetime geometry (Eq. 23).



In [Section 4.2](#), we want to interpolate between two low-energy states:  $\mathbf{x}_0^1 = (-2.55, 2.7)$  and  $\mathbf{x}_0^2 = (0.95, -0.4)$ . To avoid instabilities for  $t \approx 0$ , we represent them on the spacetime manifold as  $\boldsymbol{\theta}_1 = (-2.55, 2.7, t_{\min})$ , and  $\boldsymbol{\theta}_2 = (0.95, -0.4, t_{\min})$ , where  $\log \text{SNR}(t_{\min}) = 7$ . We then approximate the geodesic between them, by parametrizing  $\gamma$  as a cubic spline with 10 nodes and fixed endpoints  $\gamma_0 = \boldsymbol{\theta}_1$ , and  $\gamma_1 = \boldsymbol{\theta}_2$  and discretize it into 512 points. We then optimize it by minimizing [Eq. 23](#) with the Adam optimizer with learning rate  $\eta = 0.1$  and take 10000 optimization steps, which takes roughly 6 minutes on an M1 CPU.

**Annealed Langevin dynamics** To generate transition paths, we use Annealed Langevin dynamics ([Algorithm 1](#)) with the geodesic discretized into  $N = 512$  points,  $K = 128$  Langevin steps for each point on the geodesic  $\gamma$ , and use  $dt = 0.0001$ , i.e., requiring 65536 evaluations of the gradient of the auxiliary energy function. We generate 8 independent paths in parallel, which takes roughly 27 seconds on an M1 CPU.

**Constrained transition paths** Constrained transition paths were also parametrized with cubic splines with 10 nodes, but discretized into 1024 points.

For the **low-variance** transition paths, we chose the threshold  $\rho = 3$ , and  $\lambda = 0$  for the first 1200 optimization steps, and  $\lambda$  linearly increasing from 0 to 100 for the last 3800 optimization steps, for the total of 5000 optimization steps with the Adam optimizer with a learning  $\eta = 0.01$ . This took just under 6 minutes on an M1 CPU.

For the **region-avoiding** transition paths, we combine two penalty functions:  $h_1$  is the low-variance penalty described above, but with  $\rho_1 = 3.75$  threshold, and  $h_2$  is the KL penalty with  $\rho_2 = -4350$  threshold. We define  $\lambda_1$  as in the low-variance transitions, and fix  $\lambda_2 = 1$ . The optimization was performed with Adam optimizer, learning rate  $\eta = 0.1$ , and ran for 4000 steps for a runtime of just under 5 minutes on an M1 CPU.

The reason we include the low-variance penalty in the region-avoiding experiment is because  $\text{KL}(p(\cdot|\boldsymbol{\theta}^*) || p(\cdot|\boldsymbol{\gamma}_s))$  can trivially be increased by simply increasing entropy of  $p(\cdot|\boldsymbol{\gamma}_s)$  which would not result in avoiding the region defined by  $p(\cdot|\boldsymbol{\theta}^*)$ .

## F KL Fisher-Rao flow as primal and dual geodesics

On a statistical manifold, one can define the Fisher-Rao (or the natural gradient) flow as ([Müller et al., 2024](#))

$$d\boldsymbol{\theta}_s = \mathcal{I}_{\boldsymbol{\theta}_s}^{-1} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_s) ds, \quad (65)$$

where  $f$  is some objective function. It is a continuous version of the natural gradient ascent ([Martens, 2020](#)) and can be interpreted as taking infinitesimally small steps in the direction of steepest ascent of  $f$ . However, in contrast to the vanilla gradient flow, the sizes of the steps taken are measured with KL-divergence instead of the Euclidean norm.

It turns out that the Fisher-Rao flow of the KL divergence has a straight line solution in either  $\boldsymbol{\eta}$  or  $\boldsymbol{\mu}$  parametrization.

**Lemma 6** (KL Fisher-Rao flow). *Let  $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}$ . If  $\boldsymbol{\mu}$  and  $\boldsymbol{\eta}$  are invertible, then*

$$d\boldsymbol{\theta}_s = -\lambda \mathcal{I}_{\boldsymbol{\theta}_s}^{-1} \nabla_{\boldsymbol{\theta}_1} \text{KL}(\boldsymbol{\theta}_s || \boldsymbol{\theta}^*) ds \implies \boldsymbol{\eta}(\boldsymbol{\theta}_s) = (1 - e^{-\lambda s}) \boldsymbol{\eta}(\boldsymbol{\theta}^*) + e^{-\lambda s} \boldsymbol{\eta}(\boldsymbol{\theta}_0) \quad (66)$$

$$d\boldsymbol{\theta}_s = -\lambda \mathcal{I}_{\boldsymbol{\theta}_s}^{-1} \nabla_{\boldsymbol{\theta}_2} \text{KL}(\boldsymbol{\theta}^* || \boldsymbol{\theta}_s) ds \implies \boldsymbol{\mu}(\boldsymbol{\theta}_s) = (1 - e^{-\lambda s}) \boldsymbol{\mu}(\boldsymbol{\theta}^*) + e^{-\lambda s} \boldsymbol{\mu}(\boldsymbol{\theta}_0) \quad (67)$$

*Proof.* If  $\boldsymbol{\mu}$  and  $\boldsymbol{\eta}$  are invertible, we have

$$\mathcal{I}_{\boldsymbol{\theta}_s}^{-1} = \left( \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^\top \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) \right)^{-1} = \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-1} \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-\top} \quad (68)$$

The Fisher-Rao metric is always symmetric, which also implies:

$$\mathcal{I}_{\boldsymbol{\theta}_s}^{-1} = \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-1} \left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-\top}. \quad (69)$$

Now assume  $\frac{d\boldsymbol{\theta}_s}{ds} = -\lambda \mathcal{I}_{\boldsymbol{\theta}_s}^{-1} \nabla_{\boldsymbol{\theta}_1} \text{KL}(\boldsymbol{\theta}_s || \boldsymbol{\theta}^*)$ . Then, from [Lemma 3](#)

$$\begin{aligned} \frac{d}{ds} \boldsymbol{\eta}(\boldsymbol{\theta}_s) &= \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \frac{d}{ds} \boldsymbol{\theta}_s \\ &= -\lambda \underbrace{\frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \left( \frac{\partial \boldsymbol{\eta}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-1}}_{=I} \underbrace{\left( \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}} \right)^{-\top} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta}_s)}{\partial \boldsymbol{\theta}}}_{=I} (\boldsymbol{\eta}(\boldsymbol{\theta}_s) - \boldsymbol{\eta}(\boldsymbol{\theta}^*)) \\ &= -\lambda (\boldsymbol{\eta}(\boldsymbol{\theta}_s) - \boldsymbol{\eta}(\boldsymbol{\theta}^*)). \end{aligned} \quad (70)$$

Therefore, for  $f(s) = \eta(\theta_s) - \eta(\theta^*)$ , we have:

$$f'(s) = -\lambda f(s), \quad (71)$$

whose unique solution is  $f(s) = f(0)e^{-\lambda s}$ , which translates to

$$\eta(\theta_s) - \eta(\theta^*) = (\eta(\theta_0) - \eta(\theta^*)) e^{-\lambda s} \implies \eta(\theta_s) = (1 - e^{-\lambda s})\eta(\theta^*) + e^{-\lambda s}\eta(\theta_0). \quad (72)$$

Similarly for  $\frac{d\theta_s}{ds} = -\lambda \mathcal{I}_{\theta_s}^{-1} \nabla_{\theta_2} \text{KL}(\theta^* || \theta_s)$  we have

$$\begin{aligned} \frac{d}{ds} \mu(\theta_s) &= \frac{\partial \mu(\theta_s)}{\partial \theta} \frac{d}{ds} \theta_s \\ &= -\lambda \underbrace{\frac{\partial \mu(\theta_s)}{\partial \theta} \left( \frac{\partial \mu(\theta_s)}{\partial \theta} \right)^{-1}}_{=I} \underbrace{\left( \frac{\partial \eta(\theta_s)}{\partial \theta} \right)^{-\top} \frac{\partial \eta(\theta_s)}{\partial \theta}^\top}_{=I} (\mu(\theta_s) - \mu(\theta^*)) \\ &= -\lambda (\mu(\theta_s) - \mu(\theta^*)), \end{aligned} \quad (73)$$

which implies

$$\mu(\theta_s) - \mu(\theta^*) = (\mu(\theta_0) - \mu(\theta^*)) e^{-\lambda s} \implies \mu(\theta_s) = (1 - e^{-\lambda s})\mu(\theta^*) + e^{-\lambda s}\mu(\theta_0). \quad (74)$$

□

## G Expectation parameter estimation code

```

1 import jax
2 import jax.random as jr
3 import jax.numpy as jnp
4
5 def f(x, t, key): # Implemenation of the expected denoising
6     pass
7
8 def sigma_and_alpha(t): # Depends on the choice of SDE and noise schedule
9     pass
10
11 def mu(x, t, key):
12     model_key, eps_key = jr.split(key, 2)
13     eps = jr.rademacher(eps_key, (x.size,), dtype=jnp.float32)
14     def pred_fn(x_):
15         return f(x_, t, key=model_key)
16     f_pred, f_grad = jax.jvp(pred_fn, (x,), (eps,))
17     div = jnp.sum(f_grad * eps)
18     sigma, alpha = sigma_and_alpha(t)
19     return sigma**2/alpha * div + jnp.sum(f_pred ** 2), f_pred

```

Listing 1: JAX Implementation of  $\mu$  estimation

## H Additional image interpolation results

We perform an additional experiment comparing interpolation methods between images. Using the PF-ODE, we recover a clean image  $x_0$  from each point along a spacetime geodesic  $\gamma_s$ . We compare this geodesic-based interpolation to a standard baseline that interpolates in the noise space  $x_T$  using spherical linear interpolation (SLERP). While the standard interpolation, SLERP, produces sharper intermediate images, it results in significant semantic shifts. In contrast, our geodesic interpolation yields more blurred images but preserves the semantic content across the interpolation path.

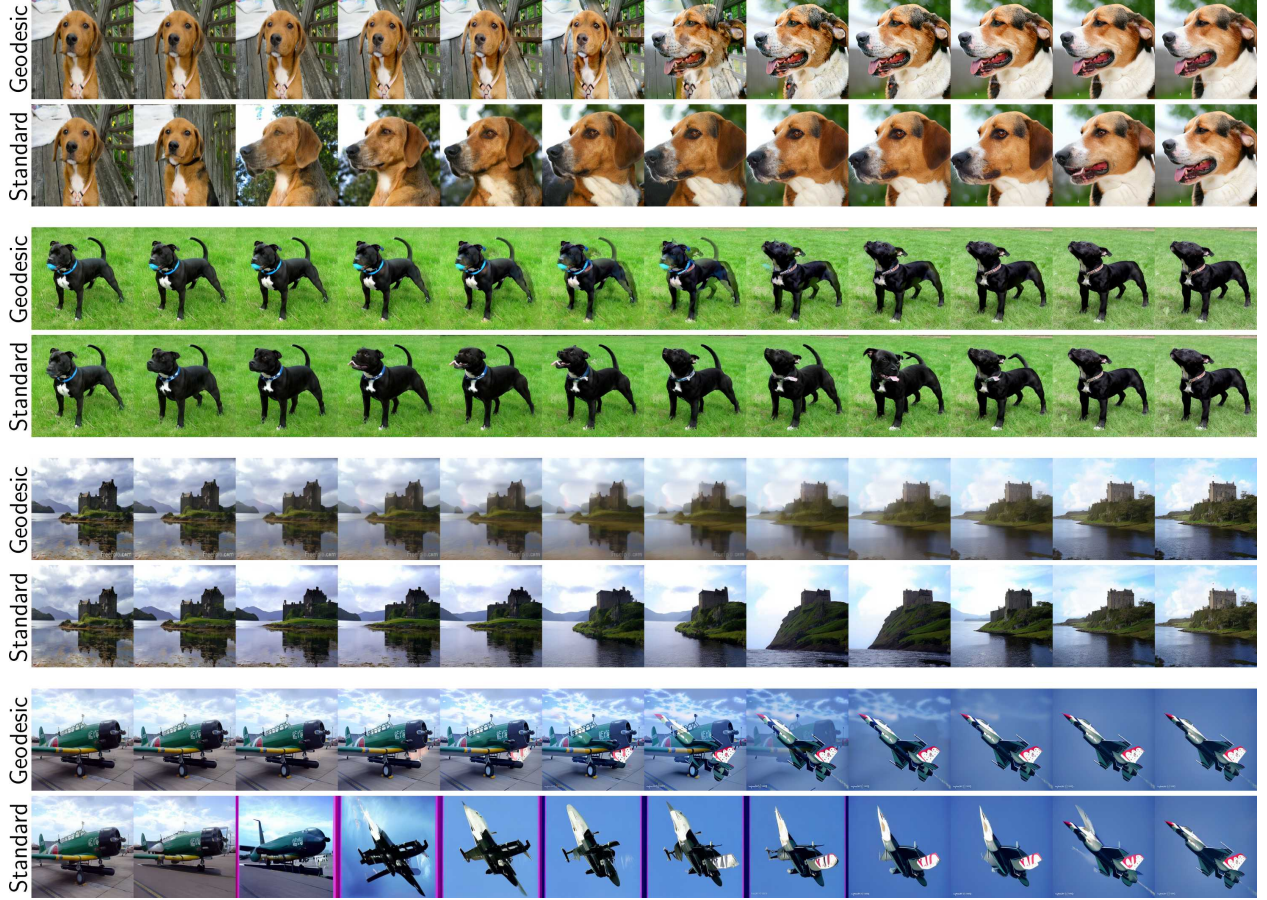


Figure 8: Spacetime geodesic decodes to less realistic images, but introduces less semantic changes than standard interpolations.

## I Broader societal impact

The use of generative models, especially those capable of producing images and videos, poses considerable risks for misuse. Such technologies have the potential to produce harmful societal effects, primarily through the spread of disinformation, but also by reinforcing harmful stereotypes and implicit biases. In this work, we contribute to a deeper understanding of diffusion models, which currently represent the leading methodology in generative modeling. While this insight may eventually support improvements to these models, thereby increasing the risk of misuse, it is important to note that our research does not introduce any new capabilities or applications of the technology.

## J Licences

- EDM2 model (Karras et al., 2024): Creative Commons BY-NC-SA 4.0 license
- ImageNet dataset (Deng et al., 2009): Custom non-commercial license
- SDVAE model (Rombach et al., 2022): CreativeML Open RAIL++-M license
- OpenM++ (OpenMP Architecture Review Board, 2008): MIT License