Journal of Machine Learning Research 26 (2025) 1-54 Submitted 1/25; Published 6/25

Autoencoders in Function Space

Justin Bunker JB2200@CANTAB.AC.UK
Department of Engineering
University of Cambridge
Cambridge, CB2 1TN, United Kingdom

Mark Girolami MGIROLAMI@TURING.AC.UK
Department of Engineering, University of Cambridge
and Alan Turing Institute
Cambridge, CB2 1TN, United Kingdom

Hefin Lambley HEFIN.LAMBLEY@WARWICK.AC.UK
Mathematics Institute
University of Warwick
Coventry, CV4 7AL, United Kingdom

Andrew M. Stuart ASTUART@CALTECH.EDU
Computing + Mathematical Sciences
California Institute of Technology
Pasadena, CA 91125, United States of America

T. J. Sullivan T.J.SULLIVAN@WARWICK.AC.UK
Mathematics Institute & School of Engineering
University of Warwick
Coventry, CV4 7AL, United Kingdom

Editor: Mahdi Soltanolkotabi

Abstract

Autoencoders have found widespread application in both their original deterministic form and in their variational formulation (VAEs). In scientific applications and in image processing it is often of interest to consider data that are viewed as functions; while discretisation (of differential equations arising in the sciences) or pixellation (of images) renders problems finite dimensional in practice, conceiving first of algorithms that operate on functions, and only then discretising or pixellating, leads to better algorithms that smoothly operate between resolutions. In this paper function-space versions of the autoencoder (FAE) and variational autoencoder (FVAE) are introduced, analysed, and deployed. Well-definedness of the objective governing VAEs is a subtle issue, particularly in function space, limiting applicability. For the FVAE objective to be well defined requires compatibility of the data distribution with the chosen generative model; this can be achieved, for example, when the data arise from a stochastic differential equation, but is generally restrictive. The FAE objective, on the other hand, is well defined in many situations where FVAE fails to be. Pairing the FVAE and FAE objectives with neural operator architectures that can be evaluated on any mesh enables new applications of autoencoders to inpainting, superresolution, and generative modelling of scientific data.

Keywords: Variational inference on function space, operator learning, variational autoencoders, regularised autoencoders, scientific machine learning

1. Introduction

Functional data, and data that can be viewed as a high-resolution approximation of functions, are ubiquitous in data science (Ramsay and Silverman, 2002). Recent years have seen much interest in machine learning in this setting, with the promise of architectures that can be trained and evaluated across resolutions. A variety of methods now exist for the supervised learning of operators between function spaces, starting with the work of Chen and Chen (1993), followed by DeepONet (Lu et al., 2021), PCA-Net (Bhattacharya et al., 2021), Fourier neural operators (FNOs; Li et al., 2021) and variants (Kovachki et al., 2023). These methods have proven useful in diverse applications such as surrogate modelling for costly simulators of dynamical systems (Azizzadenesheli et al., 2024).

Practical algorithms for functional data must necessarily operate on discrete representations of the underlying infinite-dimensional objects, identifying salient features independent of resolution. Some models make this dimension reduction explicit by representing outputs as a linear combination of basis functions—learned from data in DeepONet, and computed using principal component analysis (PCA) in PCA-Net. Others do this implicitly, as in, for example, the deep layered structure of FNOs involving repeated application of the discrete Fourier transform followed by pointwise activation.

While linear dimension-reduction methods such as PCA adapt readily to function space, there are many types of data, such as solutions to advection-dominated partial differential equations (PDEs), for which linear approximations are provably inefficient—a phenomenon known as the Kolmogorov barrier (Peherstorfer, 2022). This suggests the need for nonlinear dimension-reduction techniques on function space. Motivated by this we propose an extension of variational autoencoders (VAEs; Kingma and Welling, 2014) to functional data using operator learning; we refer to the resulting model as the **functional variational autoencoder (FVAE)**. As a probabilistic latent-variable model, FVAE allows for both dimension reduction and principled generative modelling on function space.

We define the FVAE objective as the Kullback–Leibler (KL) divergence between two joint distributions, both on the product of the data and latent spaces, defined by the encoder and decoder models; we then derive conditions under which this objective is well-defined. This differs from the usual presentation of VAEs in which one maximises a lower bound on the data likelihood, the evidence lower bound (ELBO); we show that our objective is equivalent and that it generalises naturally to function space. The FVAE objective proves to be well defined only under a compatibility condition between the data and the generative model; this condition is easily satisfied in finite dimensions but is restrictive for functional data. Our applications of FVAE rest on establishing such compatibility, which is possible for problems in the sciences such as those arising in Bayesian inverse problems with Gaussian priors (Stuart, 2010), and those governed by stochastic differential equations (SDEs) (Hairer et al., 2011). However, there are many problems in the sciences, and in generative models for PDEs in particular, for which application of FVAE fails because the generative model is incompatible with the data; using FVAE in such settings leads to foundational theoretical issues and, as a result, to practical problems in the infinite-resolution and -data limits.

To overcome these foundational issues we propose a deterministic regularised autoencoder that can be applied in very general settings, which we call the **functional autoencoder**

(FAE). We show that FAE is an effective tool for dimension reduction, and that it can used as a versatile generative model for functional data.

We complement the FVAE and FAE objectives on function space with neural-operator architectures that can be discretised on any mesh. The ability to discretise both the encoder and the decoder on arbitrary meshes extends prior work such as the variational autoencoding neural operator (VANO; Seidman et al., 2023), and is highly empowering, enabling a variety of new applications of autoencoders to scientific data such as inpainting and superresolution. Code accompanying the paper is available at

https://github.com/htlambley/functional_autoencoders.

Contributions. We make the following contributions to the development and application of autoencoders to functional data:

(C1) we propose FVAE, an extension of VAEs to function space, finding that the training objective is well defined so long as the generative model is compatible with the data;

(C2) we complement the FVAE training objective with mesh-invariant architectures that can be deployed on any discretisation—even irregular, non-grid discretisations;

(C3) we show that when the data and generative model are incompatible, the discretised FVAE objective may diverge in the infinite-resolution and -data limits or entirely fail to minimise the divergence between the encoder and decoder;

(C4) we propose FAE, an extension of regularised autoencoders to function space, and show that its objective is well defined in many cases where the FVAE objective is not;

(C5) exploiting mesh-invariance, we propose masked training schemes which exhibit greater robustness at inference time, faster training and lower memory usage;

(C6) we validate FAE and FVAE on examples from the sciences, including problems governed by SDEs and PDEs, to discover low-dimensional latent structure from data, and use our models for inpainting, superresolution, and generative modelling, exploiting the ability to discretise the encoder and decoder on any mesh.

Outline. Section 2 extends VAEs to function space (Contribution (C1)). We then describe mesh-invariant architectures of Contribution (C2); we also validate our approach, FVAE, on examples such as SDE path distributions where compatibility between the data and the generative model can be verified. Section 3 gives an example of the problems arising when applying FVAE in the "misspecified" setting of Contribution (C3). In Section 4 we propose FAE (Contribution (C4)) and apply our data-driven method to two examples from scientific machine learning: Navier–Stokes fluid flows and Darcy flows in porous media. In these problems we make use of the mesh-invariance of the FAE to apply the masked training scheme of Contribution (C5); masking proves to be vital for the applications to inpainting and superresolution in Contribution (C6). Section 5 discusses related work, and Section 6 discusses limitations and topics for future research.

2. Variational Autoencoders on Function Space

In this section we define the VAE objective by minimizing the KL divergence between two distinct joint distributions on the product of data and latent spaces, one defined by the encoder and the other by the decoder. We show that this gives a tractable objective

when written in terms of a suitable reference distribution (Section 2.1). This objective coincides with maximising the ELBO in finite dimensions (Section 2.2) and extends readily to infinite dimensions. The divergence between the encoder and decoder models is finite only under a compatibility condition between the data and the generative model. This condition is restrictive in infinite dimensions. We identify problem classes for which this compatibility holds (Section 2.3) and pair the objective with mesh-invariant encoder and decoder architectures (Section 2.4), leading to a model we call the **functional variational autoencoder (FVAE)**. We then validate our method on several problems from scientific machine learning (Section 2.5) where the data are governed by SDEs.

2.1 Training Objective

We begin by formulating an objective in infinite dimensions, making the following standard assumption in unsupervised learning throughout Section 2. At this stage we focus on the continuum problem, and address the question of discretisation in Section 2.4. In what follows $\mathscr{P}(X)$ is the set of Borel probability measures on the separable Banach space X.

Assumption 1 *Take the* ***data space*** $(\mathcal{U}, \|\cdot\|)$ *to be a separable Banach space. There exists a* ***data distribution*** $\Upsilon \in \mathscr{P}(\mathcal{U})$ *from which we have access to N independent and identically distributed samples* $\{u^{(n)}\}_{n=1}^N \subset \mathcal{U}$. ■

This setting is convenient to work with yet general enough to include many spaces of interest; $\mathcal{U}$ could be, for example, a Euclidean space $\mathbb{R}^k$, or the infinite-dimensional space $L^2(\Omega)$ of (equivalence classes of) square-integrable functions with domain $\Omega \subseteq \mathbb{R}^d$. Separability is a technical condition used to guarantee the existence of conditional distributions in the encoder and decoder models defined shortly (see Chang and Pollard, 1997, Theorem 1).

An autoencoder consists of two nonlinear transformations: an **encoder** mapping data into a low-dimensional **latent space** $\mathcal{Z}$, and a **decoder** mapping from $\mathcal{Z}$ back into $\mathcal{U}$. The goal is to choose transformations such that, for $u \sim \Upsilon$, composing the encoder and decoder approximates the identity. In this way the autoencoder can be used for dimension reduction, with the encoder output being a compressed representation of the data.

To make this precise, fix $\mathcal{Z} = \mathbb{R}^{d_\mathcal{Z}}$ and define encoder and decoder transformations depending on parameters $\theta \in \Theta$ and $\psi \in \Psi$, respectively, by

$$\text{(encoder)} \qquad \mathcal{U} \ni u \mapsto \mathbb{Q}^\theta_{z|u} \in \mathscr{P}(\mathcal{Z}), \tag{1a}$$

$$\text{(decoder)} \qquad \mathcal{Z} \ni z \mapsto \mathbb{P}^\psi_{u|z} \in \mathscr{P}(\mathcal{U}). \tag{1b}$$

To ensure the statistical models we are interested in are well defined, we require both maps to be Markov kernels—that is, for all Borel measurable sets $A \subseteq \mathcal{Z}$ and $B \subseteq \mathcal{U}$ and all parameters $\theta \in \Theta$ and $\psi \in \Psi$, the maps $u \mapsto \mathbb{Q}^\theta_{z|u}(A)$ and $z \mapsto \mathbb{P}^\psi_{u|z}(B)$ are measurable. Since the encoder and decoder both output probability distributions, we must be precise about what it means to compose them: we mean the distribution

$$\text{(autoencoding distribution)} \quad \mathbb{A}^{\theta,\psi}_u(B) = \int_\mathcal{Z} \mathbb{P}^\psi_{u|z}(B)\, \mathbb{Q}^\theta_{z|u}(\mathrm{d}z), \quad B \subseteq \mathcal{U} \text{ measurable.} \tag{2}$$

Remark 2 *Given $u \in \mathcal{U}$, $\mathbb{A}_u^{\theta,\psi}(\cdot)$ is the distribution given by drawing $z \sim \mathbb{Q}_{z|u}^\theta$ and then sampling from $\mathbb{P}_{u|z}^\psi$; we would like this to be close to a Dirac distribution δ_u when u is drawn from Υ; enforcing this, approximately, will be used to determine the parameters (θ, ψ).* ■

Autoencoding as Matching Joint Distributions. Now let us fix a **latent distribution** $\mathbb{P}_z \in \mathscr{P}(\mathcal{Z})$ and define two distributions for (z, u) on the product space $\mathcal{Z} \times \mathcal{U}$:

$$\text{(joint encoder model)} \qquad \mathbb{Q}_{z,u}^\theta(\mathrm{d}z, \mathrm{d}u) = \mathbb{Q}_{z|u}^\theta(\mathrm{d}z)\Upsilon(\mathrm{d}u), \tag{3a}$$

$$\text{(joint decoder model)} \qquad \mathbb{P}_{z,u}^\psi(\mathrm{d}z, \mathrm{d}u) = \mathbb{P}_{u|z}^\psi(\mathrm{d}u)\mathbb{P}_z(\mathrm{d}z). \tag{3b}$$

The joint encoder model (3a) is the distribution on (z, u) given by applying the encoder to data $u \sim \Upsilon$, while the joint decoder model (3b) is the distribution on (z, u) given by applying the decoder to latent vectors $z \sim \mathbb{P}_z$. We emphasise that Υ is given, while the distributions $\mathbb{Q}_{z|u}^\theta$, $\mathbb{P}_{u|z}^\psi$, and $\mathbb{P}_z$ are to be specified; the choice of decoder distribution $\mathbb{P}_{u|z}^\psi$ on the (possibly infinite-dimensional) space $\mathcal{U}$ will require particular attention in what follows.

The marginal and conditional distributions of (3a) and (3b) will be important and we write, for example, $\mathbb{P}_{z|u}^\psi$ for the $z \mid u$-conditional of the joint decoder model and $\mathbb{Q}_z^\theta$ for the z-marginal of the joint encoder model. In particular, the u-marginal of the joint decoder model is the **generative model** $\mathbb{P}_u^\psi$.

To match the joint encoder model (3a) with the joint decoder model (3b), we seek to solve, for some statistical distance or divergence d on $\mathscr{P}(\mathcal{Z} \times \mathcal{U})$, the minimisation problem

$$\operatorname*{arg\,min}_{\theta \in \Theta,\, \psi \in \Psi} \mathsf{d}\big(\mathbb{Q}_{z,u}^\theta \,\|\, \mathbb{P}_{z,u}^\psi\big). \tag{4}$$

Doing so determines an autoencoder and a generative model. The choice of d is constrained by the need for it to be possible to evaluate the objective with Υ known only empirically; examples for which this is the case include the Wasserstein metrics and the KL divergence.

VAE Objective as Minimisation of KL Divergence. Using the KL divergence as d in (4) leads to the goal of finding $\theta \in \Theta$ and $\psi \in \Psi$ to minimise

$$D_{\mathrm{KL}}\big(\mathbb{Q}_{z,u}^\theta \,\big\|\, \mathbb{P}_{z,u}^\psi\big) = \mathop{\mathbb{E}}_{(z,u)\sim\mathbb{Q}_{z,u}^\theta}\left[\log \frac{\mathrm{d}\mathbb{Q}_{z,u}^\theta}{\mathrm{d}\mathbb{P}_{z,u}^\psi}(z,u)\right] = \mathop{\mathbb{E}}_{u\sim\Upsilon}\mathop{\mathbb{E}}_{z\sim\mathbb{Q}_{z|u}^\theta}\left[\log \frac{\mathrm{d}\mathbb{Q}_{z,u}^\theta}{\mathrm{d}\mathbb{P}_{z,u}^\psi}(z,u)\right]; \tag{5}$$

here $\mathrm{d}\mathbb{Q}_{z,u}^\theta / \mathrm{d}\mathbb{P}_{z,u}^\psi$ is the Radon–Nikodym derivative of $\mathbb{Q}_{z,u}^\theta$ with respect to $\mathbb{P}_{z,u}^\psi$, the appropriate infinite-dimensional analogue of the ratio of probability densities. This exists only when $\mathbb{Q}_{z,u}^\theta$ is absolutely continuous with respect to $\mathbb{P}_{z,u}^\psi$, meaning that $\mathbb{Q}_{z,u}^\theta$ assigns probability zero to a set whenever $\mathbb{P}_{z,u}^\psi$ does; we take (5) to be infinite otherwise. This objective is equivalent to the standard VAE objective in the case $\mathcal{U} = \mathbb{R}^k$ but has the additional advantage that it can be used in the infinite-dimensional setting.

KL divergence has many benefits justifying its use across statistics. Its value in inference and generative modelling comes from the connection between maximum-likelihood methods and minimising KL divergence, as well as its information-theoretic interpretation (Cover and Thomas, 2006, Sec. 2.3). KL divergence is asymmetric and has two useful properties

justifying the order of the arguments in (5): it can be evaluated using only samples of the distribution in its first argument, and it requires no knowledge of the normalisation constant of the distribution in its second argument since minimisers of $\nu \mapsto D_{\mathrm{KL}}(\nu \,\|\, \mu)$ are invariant when scaling μ (Chen et al., 2023 and Bach et al., 2024, Sec. 12.2).

To justify the use of the joint divergence (5), in Theorem 3 we decompose the objective as the sum of two interpretable terms. In Theorem 6 we write the objective in a form that leads to actionable algorithms.

Theorem 3 *For all parameters $\theta \in \Theta$ and $\psi \in \Psi$ for which $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u} \,\|\, \mathbb{P}^\psi_{z,u}) < \infty$,*

$$D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z,u} \,\big\|\, \mathbb{P}^\psi_{z,u}\big) = \underbrace{D_{\mathrm{KL}}\big(\Upsilon \,\big\|\, \mathbb{P}^\psi_u\big)}_{\text{(I)}} + \underbrace{\mathop{\mathbb{E}}_{u\sim\Upsilon}\Big[D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}^\psi_{z|u}\big)\Big]}_{\text{(II)}}. \tag{6}$$

Proof Factorise $\mathbb{Q}^\theta_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{Q}^\theta_{z|u}(\mathrm{d}z)\Upsilon(\mathrm{d}u)$ and $\mathbb{P}^\psi_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{P}^\psi_{z|u}(\mathrm{d}z)\mathbb{P}^\psi_u(\mathrm{d}u)$ and substitute into (5) to obtain

$$D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z,u} \,\big\|\, \mathbb{P}^\psi_{z,u}\big) = \mathop{\mathbb{E}}_{\substack{(z,u)\\ \sim\mathbb{Q}^\theta_{z,u}}}\left[\log \frac{\mathrm{d}\Upsilon}{\mathrm{d}\mathbb{P}^\psi_u}(u)\frac{\mathrm{d}\mathbb{Q}^\theta_{z|u}}{\mathrm{d}\mathbb{P}^\psi_{z|u}}(z)\right] = \mathop{\mathbb{E}}_{u\sim\Upsilon}\mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\left[\log \frac{\mathrm{d}\Upsilon}{\mathrm{d}\mathbb{P}^\psi_u}(u) + \log \frac{\mathrm{d}\mathbb{Q}^\theta_{z|u}}{\mathrm{d}\mathbb{P}^\psi_{z|u}}(z)\right].$$

The result follows by inserting the definition of the KL divergences on the right-hand side of (6), and all terms are finite owing to the assumption that $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u} \,\|\, \mathbb{P}^\psi_{z,u}) < \infty$. ■

Theorem 3 decomposes the divergence (5) into (I) the error in the generative model and (II) the error in approximating the **decoder posterior** $\mathbb{P}^\psi_{z|u}$ with $\mathbb{Q}^\theta_{z|u}$ via variational inference; jointly training a variational-inference model and a generative model is exactly the goal of a VAE. However, minimising (5) makes sense only if $(\theta, \psi) \mapsto D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u} \,\|\, \mathbb{P}^\psi_{z,u})$ is finite for some θ and ψ. Verification of this property is a necessary step that we perform for our settings of interest in Propositions 8, 16, and 24.

Remark 4 (Posterior collapse) *While minimising* (5) *typically leads to a useful autoencoder, this is not guaranteed: the autoencoding distribution* (2) *may be very far from a Dirac distribution. For example, when $\mathbb{P}^\psi_{z|u} \approx \mathbb{P}_z$, the optimal decoder distribution $\mathbb{P}^\psi_{u|z}$ may well ignore the latent variable z entirely. Indeed, if $\Upsilon = \mathbb{Q}_{z|u} = \mathbb{P}_{u|z} = \mathbb{P}_z = N(0,1)$ on $\mathcal{U} = \mathbb{R}$, then $D_{\mathrm{KL}}(\mathbb{Q}_{z,u} \,\|\, \mathbb{P}_{z,u}) = 0$, but the autoencoding distribution $\mathbb{A}_u = N(0,1)$ for all $u \in \mathcal{U}$. This is not close to the desired Dirac distribution referred to in Remark 2. This issue is known in the VAE literature as **posterior collapse** (Wang et al., 2021). In practice, choice of model classes for $\mathbb{Q}_{z|u}$ and $\mathbb{P}_{u|z}$ avoids this issue.* ■

Tractable Training Objective. Since we can access Υ only through training data, we must decompose the objective function into a sum of a term which involves Υ only through samples and a term which is independent of the parameters θ and ψ. The first of these terms may then be used to define a tractable objective function over the parameters. To address this issue we introduce a **reference distribution** $\Lambda \in \mathscr{P}(\mathcal{U})$ and impose the following conditions on the encoder and the reference distribution.

Assumption 5 *(a) There exists a reference distribution $\Lambda \in \mathscr{P}(\mathcal{U})$ such that:*
(i) for all $\psi \in \Psi$ and $z \in \mathcal{Z}$, $\mathbb{P}^{\psi}_{u|z}$ is mutually absolutely continuous with Λ; and
(ii) the data distribution Υ satisfies the finite-information condition $D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda) < \infty$.
(b) For all $\theta \in \Theta$ and Υ-almost all $u \in \mathcal{U}$, $D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z|u} \,\|\, \mathbb{P}_z) < \infty$. ■

Theorem 6 *If Assumption 5 is satisfied for some $\Lambda \in \mathscr{P}(\mathcal{U})$, then for all parameters $\theta \in \Theta$ and $\psi \in \Psi$ for which $D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z,u} \,\|\, \mathbb{P}^{\psi}_{z,u}) < \infty$, we have*

$$D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta}_{z,u} \,\big\|\, \mathbb{P}^{\psi}_{z,u}\big) = \mathop{\mathbb{E}}_{u\sim\Upsilon}\big[\mathcal{L}(u;\theta,\psi)\big] + D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda), \tag{7a}$$

$$\textit{(per-sample loss)} \qquad \mathcal{L}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^{\theta}_{z|u}}\left[-\log \frac{\mathrm{d}\mathbb{P}^{\psi}_{u|z}}{\mathrm{d}\Lambda}(u)\right] + D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta}_{z|u} \,\big\|\, \mathbb{P}_z\big). \tag{7b}$$

Proof Write $\mathbb{Q}^{\theta}_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{Q}^{\theta}_{z|u}(\mathrm{d}z)\Upsilon(\mathrm{d}u)$ and $\mathbb{P}^{\psi}_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{P}^{\psi}_{u|z}(\mathrm{d}u)\mathbb{P}_z(\mathrm{d}z)$, factor through the distribution Λ in (5), and apply the definition of the KL divergence to obtain

$$\begin{aligned}
D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta}_{z,u} \,\big\|\, \mathbb{P}^{\psi}_{z,u}\big) &= \mathop{\mathbb{E}}_{(z,u)\sim\mathbb{Q}^{\theta}_{z,u}}\left[\log \frac{\mathrm{d}\mathbb{Q}^{\theta}_{z|u}}{\mathrm{d}\mathbb{P}_z}(z) \frac{\mathrm{d}\Lambda}{\mathrm{d}\mathbb{P}^{\psi}_{u|z}}(u) \frac{\mathrm{d}\Upsilon}{\mathrm{d}\Lambda}(u)\right] \\
&= \mathop{\mathbb{E}}_{u\sim\Upsilon}\left[\mathop{\mathbb{E}}_{z\sim\mathbb{Q}^{\theta}_{z|u}}\left[-\log \frac{\mathrm{d}\mathbb{P}^{\psi}_{u|z}}{\mathrm{d}\Lambda}(u) + \log \frac{\mathrm{d}\mathbb{Q}^{\theta}_{z|u}}{\mathrm{d}\mathbb{P}_z}(z)\right] + \log \frac{\mathrm{d}\Upsilon}{\mathrm{d}\Lambda}(u)\right] \\
&= \mathop{\mathbb{E}}_{u\sim\Upsilon}\left[\mathop{\mathbb{E}}_{z\sim\mathbb{Q}^{\theta}_{z|u}}\left[-\log \frac{\mathrm{d}\mathbb{P}^{\psi}_{u|z}}{\mathrm{d}\Lambda}(u)\right] + D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta}_{z|u} \,\big\|\, \mathbb{P}_z\big)\right] + D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda),
\end{aligned}$$

where all terms are finite as a consequence of Assumptions 5(a)(ii) and (b). ■

Since $D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda)$ is assumed to be finite and depends on neither θ nor ψ, Theorem 6 shows that the joint KL divergence (5) is equivalent, up to a finite constant, to

$$\text{(FVAE objective)} \qquad \mathcal{J}^{\mathrm{FVAE}}(\theta,\psi) = \mathop{\mathbb{E}}_{u\sim\Upsilon}\big[\mathcal{L}(u;\theta,\psi)\big]. \tag{8}$$

In particular, minimising $\mathcal{J}^{\mathrm{FVAE}}$ is equivalent to minimising the joint divergence, and $\mathcal{J}^{\mathrm{FVAE}}(\theta,\psi)$ is finite if and only if $D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z,u} \,\|\, \mathbb{P}^{\psi}_{z,u})$ is finite. Moreover (8) can be approximated using samples from Υ:

$$\text{(empirical FVAE objective)} \qquad \mathcal{J}^{\mathrm{FVAE}}_N(\theta,\psi) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}\big(u^{(n)};\theta,\psi\big). \tag{9}$$

In the limit of infinite data, the empirical objective (9) converges to (8); but both (8) and (9) may be infinite in many practical settings, as we shall see in the following sections.

Remark 7 *(a) Assumption 5(a) ensures that the density $\mathrm{d}\mathbb{P}^{\psi}_{u|z}/\mathrm{d}\Lambda$ in the per-sample loss exists, and that minimising* (5) *and* (8) *is equivalent; Assumption 5(b) ensures that, when the joint divergence* (5) *is finite, the per-sample loss $\mathcal{L}$ is finite for Υ-almost all $u \in \mathcal{U}$. We could also formulate Assumption 5(a) with a σ-finite reference measure Λ, e.g., Lebesgue measure, but for our theory it suffices to consider probability measures.*

(b) The proof of Theorem 6 shows why we take $\mathbb{Q}^\theta_{z,u}$ as the first argument and $\mathbb{P}^\psi_{z,u}$ as the second in (5) to obtain a tractable objective. Reversing the arguments in the divergence gives an expectation with respect to the joint decoder model (3b):

$$D_{\mathrm{KL}}\big(\mathbb{P}^\psi_{z,u} \,\big\|\, \mathbb{Q}^\theta_{z,u}\big) = \mathop{\mathbb{E}}_{(z,u)\sim\mathbb{P}^\psi_{z,u}}\left[\log \frac{\mathrm{d}\mathbb{P}_z}{\mathrm{d}\mathbb{Q}^\theta_{z|u}}(z) + \log \frac{\mathrm{d}\mathbb{P}^\psi_{u|z}}{\mathrm{d}\Lambda}(u)\right] + \mathop{\mathbb{E}}_{(z,u)\sim\mathbb{P}^\psi_{z,u}}\left[\log \frac{\mathrm{d}\Lambda}{\mathrm{d}\Upsilon}(u)\right].$$

Unlike in Theorem 6, the term involving $\mathrm{d}\Lambda/\mathrm{d}\Upsilon$ is not a constant: it depends on the parameter ψ and would therefore need to be evaluated during the optimisation process, but this is intractable because we have only samples from Υ and not its density. ■

2.2 Objective in Finite Dimensions

We now show that the theory in Section 2.1 simplifies in finite dimensions to the usual VAE objective. To do so we assume $\mathcal{U} = \mathbb{R}^k$ and that $\Upsilon \in \mathscr{P}(\mathcal{U})$ has strictly positive probability density $\upsilon\colon \mathcal{U} \to (0,\infty)$. We moreover assume that $\mathcal{Z} = \mathbb{R}^{d_\mathcal{Z}}$ and that the latent distribution and the distributions returned by the encoder (1a) and decoder (1b) are Gaussian, taking the form

$$\mathbb{Q}^\theta_{z|u} = N\big(f(u;\theta), \Sigma(u;\theta)\big) = f(u;\theta) + \Sigma(u;\theta)^{\frac{1}{2}} N(0, I_\mathcal{Z}), \tag{10a}$$

$$\mathbb{P}^\psi_{u|z} = N\big(g(z;\psi), \beta I_\mathcal{U}\big) = g(z;\psi) + \beta^{\frac{1}{2}} N\big(0, I_\mathcal{U}\big), \tag{10b}$$

$$\mathbb{P}_z = N\big(0, I_\mathcal{Z}\big), \tag{10c}$$

where $\beta > 0$ is fixed and the parameters of (10a) and (10b) are given by learnable maps

$$\mathsf{f} = (f, \Sigma)\colon \mathcal{U} \times \Theta \to \mathcal{Z} \times \mathcal{S}_+(\mathcal{Z}), \tag{11a}$$

$$g\colon \mathcal{Z} \times \Psi \to \mathcal{U}, \tag{11b}$$

with $\mathcal{S}_+(\mathcal{Z})$ denoting the set of positive semidefinite matrices on $\mathcal{Z}$. The Gaussian model (10a)–(10c) is the standard setting in which VAEs are applied, resulting in the joint decoder model (3b) being the distribution of (z,u) in the model

$$u \mid z = g(z;\psi) + \eta, \qquad z \sim N\big(0, I_\mathcal{Z}\big), \qquad \eta \sim \mathbb{P}_\eta = \beta^{\frac{1}{2}} N\big(0, I_\mathcal{U}\big). \tag{12}$$

Other decoder models are also possible (Kingma and Welling, 2014), and in infinite dimensions we will consider a wide class of decoders, including Gaussians as a particular case. In practice (11a), (11b) will come from a parametrised class of functions, e.g., a class of neural networks. Provided these classes are sufficiently large and the data distribution has finite information with respect to $\mathbb{P}_\eta$, the joint divergence (5) is finite for at least one choice of θ and ψ.

Proposition 8 *Suppose that for parameters $\theta^\star \in \Theta$ and $\psi^\star \in \Psi$ we have $\mathsf{f}(u;\theta^\star) = (0, I_\mathcal{Z})$ and $g(z;\psi^\star) = 0$. Then*

$$D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta^\star}_{z,u} \,\big\|\, \mathbb{P}^{\psi^\star}_{z,u}\big) = D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta).$$

In particular, if $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$, then the joint divergence (5) has finite infimum.

Proof Evaluating the joint encoder model (3a) and the joint decoder model (3b) at $\theta^\star$ and $\psi^\star$ gives $\mathbb{Q}^{\theta^\star}_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{P}_z(\mathrm{d}z)\Upsilon(\mathrm{d}u)$ and $\mathbb{P}^{\psi^\star}_{z,u}(\mathrm{d}z, \mathrm{d}u) = \mathbb{P}_z(\mathrm{d}z)\mathbb{P}_\eta(\mathrm{d}u)$, so

$$D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta^\star}_{z,u} \,\big\|\, \mathbb{P}^{\psi^\star}_{z,u}\big) = \mathop{\mathbb{E}}_{u\sim\Upsilon} \mathop{\mathbb{E}}_{z\sim\mathbb{P}_z} \left[\log \frac{\mathrm{d}\mathbb{P}_z}{\mathrm{d}\mathbb{P}_z}(z) + \log \frac{\mathrm{d}\Upsilon}{\mathrm{d}\mathbb{P}_\eta}(u)\right] = D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta). \qquad \blacksquare$$

Remark 9 *In finite dimensions, many data distributions satisfy $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$. However there are cases in which this fails, e.g., when Υ is a Cauchy distribution on $\mathbb{R}$, In such cases* (5) *is infinite even with the trivial maps $\mathsf{f}(u;\theta^\star) = (0, I_{\mathcal{Z}})$ and $g(z;\psi^\star) = 0$.* $\blacksquare$

Training Objective. In Section 2.1 we proved that the joint divergence (5) can be approximated by the tractable objective (8) and its empiricalisation (9) whenever there is a reference distribution Λ satisfying Assumption 5. We now show that taking $\Lambda = \Upsilon$ satisfies this assumption and results in a training objective equivalent to maximising the ELBO. Recall the data density υ associated with data measure Υ.

Proposition 10 *Under the model* (10a)–(10c) *with reference distribution $\Lambda = \Upsilon$, Assumption 5 is satisfied, and, for some finite constant $C > 0$ independent of u, θ, and ψ,*

$$\mathcal{L}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}} \big[-\log p^\psi_{u|z}(u)\big] + \log \upsilon(u) + D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}_z\big) \tag{13a}$$

$$= \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}} \Big[(2\beta)^{-1}\big\|g(z;\psi) - u\big\|_2^2\Big] + \log \upsilon(u) + D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}_z\big) + C. \tag{13b}$$

Proof For Assumption 5(a), mutual absolute continuity of $\mathbb{P}^\psi_{u|z}$ and Λ follows as both distributions have strictly positive densities, and evidently $D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda) = 0$. Assumption 5(b) holds since both the encoder distribution (10a) and the latent distribution (10c) are Gaussian, with KL divergence available in closed form (Remark 11). The expression for $\mathcal{L}$ follows from (7b) using that $\mathbb{P}^\psi_{u|z}$ is Gaussian with density $p^\psi_{u|z}$ and $\mathrm{d}\mathbb{P}^\psi_{u|z}/\mathrm{d}\Upsilon(u) = p^\psi_{u|z}(u)/\upsilon(u)$. $\blacksquare$

While the per-sample loss (13a) involves the unknown density υ, we can drop this without affecting the objective (8) if Υ has finite **differential entropy** $\mathbb{E}_{u\sim\Upsilon}\big[-\log \upsilon(u)\big]$. This follows because

$$\mathcal{J}^{\mathrm{FVAE}}(\theta,\psi) = \mathop{\mathbb{E}}_{u\sim\Upsilon}\big[\mathcal{L}^{\mathrm{VAE}}(u;\theta,\psi)\big] - \mathop{\mathbb{E}}_{u\sim\Upsilon}\big[-\log \upsilon(u)\big] + C, \tag{14a}$$

$$\mathcal{L}^{\mathrm{VAE}}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}} \Big[(2\beta)^{-1}\big\|g(z;\psi) - u\big\|_2^2\Big] + D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}_z\big). \tag{14b}$$

Thus, $\mathcal{J}^{\mathrm{VAE}}(\theta,\psi) = \mathbb{E}_{u\sim\Upsilon}[\mathcal{L}^{\mathrm{VAE}}(\theta,\psi)]$ is equivalent, up to a finite constant, to $\mathcal{J}^{\mathrm{FVAE}}(\theta,\psi)$, and $\mathcal{J}^{\mathrm{VAE}}$ is tractable. Requiring that Υ has finite differential entropy is a mild condition, and one can expect this to be the case for the vast majority of distributions arising in the finite-dimensional setting.

Remark 11 (VAEs as regularised autoencoders) *We can write the divergence in* (14b) *in closed form as* $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z|u} \,\|\, \mathbb{P}_z) = \frac{1}{2}\big(\|f(u;\theta)\|_2^2 + \operatorname{tr}\big(\Sigma(u;\theta) - \log\Sigma(u;\theta)\big) - d_{\mathcal{Z}}\big)$*. Applying the reparametrisation trick (Kingma and Welling, 2014) to write the expectation over* $z \sim \mathbb{Q}^\theta_{z|u}$ *in terms of* $\xi \sim N(0, I_{\mathcal{Z}})$*, the interpretation of a VAE as a regularised autoencoder is clear:*

$$\begin{aligned}\mathcal{L}^{VAE}(u;\theta,\psi) \propto \mathop{\mathbb{E}}_{\xi\sim N(0,I_{\mathcal{Z}})}\Big[(2\beta)^{-1}\Big\|g\Big(f(u;\theta)+\Sigma(u;\theta)\xi;\psi\Big)-u\Big\|_2^2\Big] \\ + \tfrac{1}{2}\,\|f(u;\theta)\|_2^2 + \tfrac{1}{2}\operatorname{tr}\big(\Sigma(u;\theta)-\log\Sigma(u;\theta)\big).\end{aligned}$$

■

Remark 12 (The evidence lower bound) *The usual derivation of VAEs specifies the decoder model* (12) *and performs variational inference on the posterior for* $z \mid u$*, seeking to maximise a lower bound, the ELBO, on the likelihood of the data. Denoting by* p^ψ_u *the density of the generative model* $\mathbb{P}^\psi_u$*, by* $q^\theta_{z|u}$ *the density of* $\mathbb{Q}^\theta_{z|u}$*, and so forth, the log-likelihood of* $u \in \mathcal{U}$ *is*

$$\log p^\psi_u(u) = D_{\mathrm{KL}}\big(q^\theta_{z|u} \,\big\|\, p^\psi_{z|u}\big) + \mathrm{ELBO}(u;\theta,\psi), \tag{15a}$$

$$\mathrm{ELBO}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim q^\theta_{z|u}}\big[\log p^\psi_{u|z}(u)\big] - D_{\mathrm{KL}}\big(q^\theta_{z|u} \,\big\|\, p_z\big). \tag{15b}$$

This demonstrates that $\mathrm{ELBO}(u;\theta,\psi)$ *is indeed a lower bound on the log-data likelihood under the decoder model. Minimising our per-sample loss* (13a) *is equivalent to maximising the ELBO* (15b)*, but, notably, our underlying objective is shown to correspond exactly to the underlying KL divergence* (5) *and avoids the use of any bounds on data likelihood.* ■

Remark 13 *An interpretation of the VAE objective as minimisation of* (5) *is also adopted by Kingma (2017, Sec. 2.8) and Kingma and Welling (2019). Our approach differs by writing the joint decoder model* (3b) *in terms of* Υ *rather than the empirical distribution*

$$\textit{(empirical data distribution)} \qquad \Upsilon_N = \frac{1}{N}\sum_{n=1}^{N}\delta_{u^{(n)}}.$$

Since Υ_N *is not absolutely continuous with respect to the generative model* $\mathbb{P}^\psi_u$ *of* (10a)–(10c)*, using this would result in the joint divergence* (5) *being infinite for all* θ *and* ψ*.* ■

2.3 Objective in Infinite Dimensions

We return to the setting of Assumption 1 and adopt a generalisation of the Gaussian model (10a)–(10c), with distributional parameters given by learnable maps $\mathsf{f} = (f, \Sigma)\colon \mathcal{U}\times\Theta \to \mathcal{Z}\times\mathcal{S}_+(\mathcal{Z})$ and $g\colon \mathcal{Z}\times\Psi\to\mathcal{U}$:

$$\mathbb{Q}^\theta_{z|u} = N\big(f(u;\theta),\Sigma(u;\theta)\big) = f(u;\theta) + \Sigma(u;\theta)^{\frac{1}{2}}N(0,I_{\mathcal{Z}}), \tag{16a}$$

$$\mathbb{P}^\psi_{u|z} = \mathbb{P}_\eta\big(\cdot - g(z;\psi)\big) = g(z;\psi) + \mathbb{P}_\eta, \tag{16b}$$

$$\mathbb{P}_\eta \in \mathscr{P}(\mathcal{U}), \tag{16c}$$

$$\mathbb{P}_z = N\big(0, I_{\mathcal{Z}}\big). \tag{16d}$$

The only change from the finite-dimensional model is in the decoder distribution (16b), which we now write as the shift of the **decoder-noise distribution** $\mathbb{P}_\eta$ by the mean $g(z;\psi)$. As we will see shortly, the choice of decoder noise will be very important in infinite dimensions, and restricting attention solely to Gaussian white noise will no longer be feasible.

Obstacles in Infinite Dimensions. Many new issues arise in infinite dimensions, necessitating a more careful treatment of the FVAE objective; for example, we can no longer work with probability density functions, since there is no uniform measure analogous to Lebesgue measure (Sudakov, 1959). The fundamental obstacle, however, is that—unlike in finite dimensions—the joint divergence (5) is often ill defined, satisfying

$$D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z,u} \,\big\|\, \mathbb{P}^\psi_{z,u}\big) = \infty \quad \text{for all } \theta \in \Theta \text{ and } \psi \in \Psi.$$

An important situation in which this arises is misspecification of the generative model $\mathbb{P}^\psi_u$; consequently, great care is needed in the choice of decoder to avoid this issue. To illustrate this we show that the extension of the white-noise model of (10a)–(10c) is ill-defined in infinite dimensions: the resulting joint divergence (5) is always infinite. To do this we define white noise using a Karhunen–Loève expansion (Sullivan, 2015, Sec. 11.1).

Definition 14 *Let $\mathcal{U} = L^2([0,1])$ with orthonormal basis $e_j(x) = \sqrt{2}\sin(\pi j x)$, $j \in \mathbb{N}$. We say that the random variable η is L^2-**white noise** if it has Karhunen–Loève expansion*

$$(L^2\text{-white noise}) \qquad \eta = \sum_{j\in\mathbb{N}} \xi_j e_j, \qquad \xi_j \overset{i.i.d.}{\sim} N(0,1). \qquad \blacksquare$$

Proposition 15 *Let $\mathbb{P}_\eta$ be the distribution of the L^2-white noise η. Then $\mathbb{P}_\eta$ is a probability distribution supported on the Sobolev space $H^s([0,1])$ if and only if $s < -1/2$; in particular $\mathbb{P}_\eta$ assigns probability zero to $L^2([0,1])$. Moreover, these statements remain true for any shift $\mathbb{P}_\eta(\cdot - h)$ of the distribution $\mathbb{P}_\eta$ by $h \in L^2([0,1])$.*

The proof, which makes use of the Borel–Cantelli lemma and the Kolmogorov two-series theorem, is stated in Appendix A.

Example 1 *Let $\mathcal{U} = L^2([0,1])$, fix a data distribution $\Upsilon \in \mathscr{P}(\mathcal{U})$ and take the model (16a)–(16d), where we further assume that $g(z;\psi) \in L^2([0,1])$ for all $z \in \mathcal{Z}$ and $\psi \in \Psi$, and with $\mathbb{P}_\eta$ taken to be the distribution of L^2-white noise. Under this model, the joint divergence (5) is infinite for all θ and ψ. To see this, note that Υ assigns probability one to $\mathcal{U}$, but, as $\mathbb{P}^\psi_{u|z}$ assigns zero probability to $\mathcal{U}$ by Proposition 15, we have*

$$\mathbb{P}^\psi_u(\mathcal{U}) = \int_{\mathcal{Z}} \mathbb{P}^\psi_{u|z}(\mathcal{U})\, \mathbb{P}_z(\mathrm{d}z) = 0.$$

Thus Υ is not absolutely continuous with respect to $\mathbb{P}^\psi_u$, and so $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u} \,\|\, \mathbb{P}^\psi_{z,u}) = \infty$. $\blacksquare$

The VANO model (Seidman et al., 2023), which we discuss in detail in the related work (Section 5), adopts the setting of (16a)–(16d) with white decoder noise. It thus suffers from not being well-defined. The issues in Example 1 stem from the difference in regularity

between the data, which lies in $L^2([0,1])$, and draws from the generative model, which lie in $H^s([0,1])$, $s < -1/2$, and not in $L^2([0,1])$. A difference in regularity is not the only possible issue: even two Gaussians supported on the same space need not be absolutely continuous owing to the Feldman–Hájek theorem (Bogachev, 1998, Ex. 2.7.4). But, as in Proposition 8, we can state a sufficient condition for the divergence (5) to have finite infimum.

Proposition 16 *Suppose that* $\mathsf{f}(u;\theta^\star) = (0, I_{\mathcal{Z}})$ *and* $g(z;\psi^\star) = 0$ *for some* $\theta^\star \in \Theta$ *and* $\psi^\star \in \Psi$*, and that* $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$*. Then* (5) *has finite infimum.*

Well-Defined Objective for Specific Problem Classes. The issues we have seen suggest that one must choose the decoder model based on the structure of the data distribution: fixing a decoder model *a priori* typically results, in infinite dimensions, in the joint divergence being infinite. However we now give examples to show that, in important classes of problems arising in science and engineering, there is a clear choice of decoder noise $\mathbb{P}_\eta$ arising from the problem structure. Adopting this decoder noise and taking the reference distribution $\Lambda = \mathbb{P}_\eta$ will ensure that the hypotheses of Theorem 6 are satisfied: the joint divergence can be shown to have finite infimum, and Assumption 5 holds. Thus we can apply the actionable algorithms derived in Section 2.1.

2.3.1 SDE Path Distributions

One class of data to which we can apply FVAE arises in the study of random dynamical systems (E et al., 2004). We choose Υ to be the distribution over paths of the diffusion process defined, for a standard Brownian motion $(w_t)_{t\in[0,T]}$ on $\mathbb{R}^m$ and $\varepsilon > 0$, by

$$\mathrm{d}u_t = b(u_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \qquad u_0 = 0, \qquad t \in [0,T]. \tag{17}$$

We assume that the drift $b\colon \mathbb{R}^m \to \mathbb{R}^m$ is regular enough that (17) is well defined. The theory we outline also applies to systems with anisotropic diffusion and time-dependent coefficients (Särkkä and Solin, 2019, Sec. 7.3), and to systems with nonzero initial condition, but we focus on the setting (17) for simplicity. The path distribution Υ is defined on the space $\mathcal{U} = C_0([0,T],\mathbb{R}^m)$ of continuous functions $u\colon [0,T] \to \mathbb{R}^m$ with $u(0) = 0$.

Recall (16b) where we define the decoder distribution $\mathbb{P}^\psi_{u|z}$ as the shift of the noise distribution $\mathbb{P}_\eta$ by $g(z;\psi)$, and recall Assumption 5, which in particular demands that $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$ and that $\mathbb{P}^\psi_{u|z}$ is mutually absolutely continuous with $\mathbb{P}_\eta$. Here we choose $\mathbb{P}_\eta$ to be the law of an auxiliary diffusion process, which in our examples will be an Ornstein–Uhlenbeck (OU) process. This auxiliary process must have zero initial condition and must have the same noise structure as that in (17) to ensure that $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$. We will learn the shift g and this will have to satisfy the zero initial condition to ensure that $\mathbb{P}^\psi_{u|z}$ is mutually absolutely continuous with $\mathbb{P}_\eta$.

Decoder-Noise Distribution $\mathbb{P}_\eta$. We take $\mathbb{P}_\eta$ to be the law of the auxiliary SDE

$$\mathrm{d}\eta_t = c(\eta_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \qquad \eta_0 = 0, \qquad t \in [0,T], \tag{18}$$

with drift $c\colon \mathbb{R}^m \to \mathbb{R}^m$, and with $\varepsilon > 0$ being the same in both (17) and (18). To determine whether $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$, we will need to evaluate the term $\mathrm{d}\Upsilon/\mathrm{d}\mathbb{P}_\eta$ in the KL divergence; an expression for this is given by the Girsanov theorem (Liptser and Shiryaev, 2001, Chap. 7).

Proposition 17 (Girsanov theorem) *Suppose that $\mathcal{U} = C_0([0,T],\mathbb{R}^m)$ and that $\mu \in \mathscr{P}(\mathcal{U})$ and $\nu \in \mathscr{P}(\mathcal{U})$ are the laws of the $\mathbb{R}^m$-valued diffusions*

$$\begin{aligned} \mathrm{d}u_t &= p(u_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \quad u_0 = 0, \quad t \in [0,T], \\ \mathrm{d}v_t &= q(v_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \quad v_0 = 0, \quad t \in [0,T]. \end{aligned}$$

Suppose that the Novikov condition (Øksendal, 2003, eq. (8.6.8)) holds for both processes:

$$\mathop{\mathbb{E}}_{u\sim\mu}\left[\int_0^T \|p(u_t)\|_2^2\,\mathrm{d}t\right] < \infty \quad \textit{and} \quad \mathop{\mathbb{E}}_{v\sim\nu}\left[\int_0^T \|q(v_t)\|_2^2\,\mathrm{d}t\right] < \infty. \tag{19}$$

Then

$$\frac{\mathrm{d}\mu}{\mathrm{d}\nu}(u) = \exp\left(\frac{1}{2\varepsilon}\int_0^T \|q(u_t)\|_2^2 - \|p(u_t)\|_2^2\,\mathrm{d}t - \frac{1}{\varepsilon}\int_0^T \langle q(u_t) - p(u_t),\, \mathrm{d}u_t\rangle\right). \tag{20}$$

The second integral in (20) is a **stochastic integral** with respect to $(u_t)_{t\in[0,T]}$ (Särkkä and Solin, 2019, Chap. 4). The Novikov condition suffices for our needs, but the theorem also holds under weaker conditions such as the Kazamaki condition (Liptser and Shiryaev, 2001, p. 249). Applying Proposition 17 to evaluate the KL divergence (Appendix A) yields

$$D_{\mathrm{KL}}\big(\Upsilon \,\big\|\, \mathbb{P}_\eta\big) = \mathop{\mathbb{E}}_{u\sim\Upsilon}\left[\frac{1}{2\varepsilon}\int_0^T \|b(u_t) - c(u_t)\|_2^2\,\mathrm{d}t\right].$$

Thus the condition $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$ is satisfied quite broadly, e.g., if b and c are bounded.

Per-Sample Loss. With the condition $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_\eta) < \infty$ verified, it remains to choose g to ensure that $\mathbb{P}^\psi_{u|z}$ is mutually absolutely continuous with $\mathbb{P}_\eta$, and to derive the corresponding density. Once this is done, we arrive at the actionable per-sample loss derived in Theorem 6. To do this we again apply the Girsanov theorem, using that, when $g(z;\psi) \in H^1([0,T])$ takes value zero at $t = 0$, the distribution $\mathbb{P}^\psi_{u|z}$ is the law of the SDE

$$\mathrm{d}v_t = g(z;\psi)'(t)\,\mathrm{d}t + c\big(v_t - g(z;\psi)\big)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \quad v_0 = 0, \quad t \in [0,T]. \tag{21}$$

As we will parametrise $g(z;\psi)$ using a neural network, we can assume it to be in $C^1([0,T])$, and hence in $H^1([0,T])$; moreover we shall enforce that $g(z;\psi)(0) \approx 0$ through the loss, as described shortly. To be concrete, we restrict attention to decoder noise with $c(x) = -\kappa x$, making $(\eta_t)_{t\in[0,T]}$ an OU process if $\kappa > 0$ and Brownian motion if $\kappa = 0$. In this case the per-sample loss can be derived by applying the Girsanov theorem to (18) and (21).

Proposition 18 (SDE per-sample loss) *Suppose that $c(x) = -\kappa x$, $\kappa \geqslant 0$, and that $g(z;\psi) \in H^1([0,T])$ with $g(z;\psi)(0) = 0$ for all $z \in \mathcal{Z}$ and $\psi \in \Psi$. Then Assumption 5 holds and*

$$\begin{aligned} \mathcal{L}(u;\theta,\psi) &= \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\left[-\log\frac{\mathrm{d}\mathbb{P}^\psi_{u|z}}{\mathrm{d}\mathbb{P}_\eta}(u)\right] + D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z|u} \,\|\, \mathbb{P}_z), \\ \log\frac{\mathrm{d}\mathbb{P}^\psi_{u|z}}{\mathrm{d}\mathbb{P}_\eta}(u) &= \frac{1}{\varepsilon}\int_0^T \big\langle g(z;\psi)'(t) + \kappa g(z;\psi)(t), \mathrm{d}u_t\big\rangle \\ &\qquad -\frac{1}{2\varepsilon}\int_0^T \Big(\big\|g(z;\psi)'(t) - \kappa\big(u(t) - g(z;\psi)(t)\big)\big\|_2^2 - \|\kappa u(t)\|_2^2\Big)\,\mathrm{d}t. \end{aligned}$$

In practice, we make two modifications to $\mathcal{L}$. First, the initial condition $g(z;\psi)(0) = 0$ is not enforced exactly; instead, we add a Tikhonov-like zero-penalty term with regularisation parameter $\lambda > 0$ to favour $g(z;\psi)(0) \approx 0$. Second, to allow variation of the strength of the KL regularisation, we multiply the term $D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z|u} \,\|\, \mathbb{P}_z)$ by a regularisation parameter $\beta > 0$. Setting $\beta \neq 1$ breaks the exact correspondence between the FVAE objective and the joint KL divergence (5), but can nevertheless be useful in computational practice (Higgins et al., 2017). This leads us to the SDE per-sample loss

$$\mathcal{L}^{\mathrm{SDE}}_{\lambda,\beta}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^{\theta}_{z|u}}\left[-\log \frac{\mathrm{d}\mathbb{P}^{\psi}_{u|z}}{\mathrm{d}\mathbb{P}_{\eta}}(u) + \lambda\|g(z;\psi)(0)\|_2^2\right] + \beta D_{\mathrm{KL}}\big(\mathbb{Q}^{\theta}_{z|u} \,\big\|\, \mathbb{P}_z\big).$$

2.3.2 Posterior Distributions in Bayesian Inverse Problems

Our theory can also be applied to to posterior distributions arising in Bayesian inverse problems (Stuart, 2010), which we illustrate through the following additive-noise inverse problem. Let $\mathcal{U}$ be a separable Hilbert space with norm $\|\cdot\|_{\mathcal{U}}$, let $Y = \mathbb{R}^{d_Y}$, and let $\mathcal{G}\colon \mathcal{U} \to Y$ be a (possibly nonlinear) observation operator. Suppose that $y \in Y$ is given by the model

$$y = \mathcal{G}(u) + \xi, \qquad u \sim \mu_0 \in \mathscr{P}(\mathcal{U}), \qquad \xi \sim N(0,\Sigma) \in \mathscr{P}(Y), \tag{22}$$

with noise covariance $\Sigma \in \mathcal{S}_+(Y)$, and with prior distribution $\mu_0 = N(0,C)$ having covariance operator $C\colon \mathcal{U} \to \mathcal{U}$. Models of this type arise, for example, in both Eulerian and Lagrangian data assimilation problems in oceanography (Cotter et al., 2010). Given an observation $y \in Y$ from (22), the Bayesian approach seeks to infer $u \in \mathcal{U}$ by computing the posterior distribution $\mu^y \in \mathscr{P}(\mathcal{U})$ representing the distribution of $u \mid y$. In the setting of (22), μ^y has a density with respect to μ_0 thanks to Bayes' rule (Dashti and Stuart, 2017, Theorem 14), taking the form

$$\frac{\mathrm{d}\mu^y}{\mathrm{d}\mu_0}(u) = \frac{1}{Z(y)}\exp\big(-\Phi(u;y)\big), \quad \Phi(u;y) = \frac{1}{2}\|\mathcal{G}(u) - y\|^2_{\Sigma}, \quad \|\cdot\|_{\Sigma} = \|\Sigma^{-1/2}\cdot\|_2,$$

where $Z(y) \in (0,1]$ owing to the nonnegativity of Φ. A simple calculation then reveals

$$D_{\mathrm{KL}}(\mu^y \,\|\, \mu_0) = \mathop{\mathbb{E}}_{u\sim\mu^y}\left[\log\frac{\mathrm{d}\mu^y}{\mathrm{d}\mu_0}(u)\right] = \mathop{\mathbb{E}}_{u\sim\Upsilon}\big[-\log Z(y) - \Phi(u)\big] \leqslant -\log Z(y) < \infty. \tag{23}$$

Similar arguments apply quite generally for observation models other than (22), provided the resulting log-density $\log \mathrm{d}\mu^y/\mathrm{d}\mu_0$ satisfies suitable boundedness or integrability conditions.

We now assume that the data distribution Υ to be learned is the posterior μ^y, and that we have samples from it. This setting could arise, for example, when attempting to generate further approximate samples from the posterior μ^y, taking as data the output of a function-space MCMC method (Cotter et al., 2013), with the ambition of faster sampling under FVAE than under MCMC. Recall the definition of the decoder distribution $\mathbb{P}^{\psi}_{u|z}$ in (16b). We take $\mathbb{P}_{\eta}$ to be the prior μ_0; this is a natural choice as (23) shows that $D_{\mathrm{KL}}(\Upsilon \,\|\, \mathbb{P}_{\eta}) < \infty$. We next discuss the choice of shift g, and the per-sample loss that results from these choices.

Per-Sample Loss. Since $\mathbb{P}_\eta$ and $\mathbb{P}^\psi_{u|z}$ are Gaussian, we can use the Cameron–Martin theorem (Bogachev, 1998, Corollary 2.4.3) to derive conditions for their mutual absolute continuity. For this to be the case, the shift $g(z;\psi)$ must lie in the **Cameron–Martin space** $H(\mathbb{P}_\eta) \subset \mathcal{U}$. Before stating the theorem, we recall the following facts about Gaussian measures. The space $H(\mathbb{P}_\eta)$ is Hilbert, and for fixed $h \in H(\mathbb{P}_\eta)$, the $H(\mathbb{P}_\eta)$-inner product $\langle h, \cdot \rangle_{H(\mathbb{P}_\eta)}$ extends uniquely (up to equivalence $\mathbb{P}_\eta$-almost everywhere) to a **measurable linear functional** (Bogachev, 1998, Theorem 2.10.11), denoted by $\mathcal{U} \ni u \mapsto \langle h, u \rangle^\sim_{H(\mathbb{P}_\eta)}$.

Proposition 19 (Cameron–Martin theorem) *Let $\mathbb{P}_\eta \in \mathscr{P}(\mathcal{U})$ be a Gaussian measure with Cameron–Martin space $H(\mathbb{P}_\eta)$. Then $\mathbb{P}^\psi_{u|z} = \mathbb{P}_\eta\big(\cdot - g(z;\psi)\big)$ is mutually absolutely continuous with $\mathbb{P}_\eta$ if and only if $g(z;\psi) \in H(\mathbb{P}_\eta)$, and*

$$\frac{\mathrm{d}\mathbb{P}^\psi_{u|z}}{\mathrm{d}\mathbb{P}_\eta}(u) = \exp\Big(\langle g(z;\psi), u\rangle^\sim_{H(\mathbb{P}_\eta)} - \tfrac{1}{2}\|g(z;\psi)\|^2_{H(\mathbb{P}_\eta)}\Big). \tag{24}$$

Remark 20 *The exponent in (24) should be viewed as the misfit $\frac{1}{2}\|g(z;\psi) - u\|^2_{H(\mathbb{P}_\eta)}$ with the almost-surely-infinite term $\frac{1}{2}\|u\|^2_{H(\mathbb{P}_\eta)}$ subtracted (Stuart, 2010, Remark 3.8). When $\mathbb{P}_\eta$ is Brownian motion on $\mathbb{R}$, for example, $\langle g(z;\psi), u\rangle^\sim_{H(\mathbb{P}_\eta)}$ is a stochastic integral and $H(\mathbb{P}_\eta) = H^1([0,T])$; this is implicit in the calculations underlying Theorem 18. When $\mathbb{P}_\eta$ is L^2-white noise, $H(\mathbb{P}_\eta) = L^2([0,1])$.* ■

When g takes values in $H(\mathbb{P}_\eta)$, we can use the Cameron–Martin theorem to write down the per-sample loss explicitly since $H(\mathbb{P}_\eta) = C^{1/2}\mathcal{U}$, with $\|h\|_{H(\mathbb{P}_\eta)} = \|C^{-1/2}h\|_{\mathcal{U}}$ and $\langle h, u\rangle^\sim_{H(\mathbb{P}_\eta)} = \langle C^{-1/2}h, C^{-1/2}u\rangle_{\mathcal{U}}$ for $h \in H(\mathbb{P}_\eta)$ and $u \in \mathcal{U}$.

Proposition 21 (Bayesian inverse problem per-sample loss) *Suppose that $g(z;\psi) \in H(\mathbb{P}_\eta)$ for all $z \in \mathcal{Z}$ and $\psi \in \Psi$. Then Assumption 5 holds and*

$$\mathcal{L}^{BIP}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\Big[\tfrac{1}{2}\big\|C^{-\frac{1}{2}}g(z;\psi)\big\|^2_{\mathcal{U}} - \big\langle C^{-\frac{1}{2}}g(z;\psi), C^{-\frac{1}{2}}u\big\rangle_{\mathcal{U}}\Big] + D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u}\,\big\|\,\mathbb{P}_z\big).$$

Depending on the choice of $\mathbb{P}_\eta$, the condition $g(z;\psi) \in H(\mathbb{P}_\eta)$ may follow immediately, e.g., when $\mathbb{P}_\eta$ is Brownian motion and g is parametrised by a neural network.

2.4 Architecture and Algorithms

In practice, we do not have access to training data $\{u^{(n)}\}_{n=1}^N \subset \mathcal{U}$; instead we have access to finite-dimensional discretisations $\boldsymbol{u}^{(n)}$. We would like to evaluate the encoder and decoder, and to compute the empirical objective (9) for training, using only these discretisations.

In our architectures we will assume that $\mathcal{U}$ is a Banach space of functions evaluable pointwise almost everywhere with domain $\Omega \subseteq \mathbb{R}^d$ and range $\mathbb{R}^m$; in this setting we will assume that the discretisation $\boldsymbol{u}$ of a function $u \in \mathcal{U}$ consists of evaluations at $\{x_i\}_{i=1}^I \subset \Omega$:

$$\text{(discretisation of function } u \in \mathcal{U}) \qquad \boldsymbol{u} = \Big\{\Big(x_i, u(x_i)\Big)\Big\}_{i=1}^I \subset \Omega \times \mathbb{R}^m.$$

Crucially, the number and location of mesh points may differ for each discretised sample—our aim is to allow for FVAE to be trained and evaluated across different resolutions, with data provided on sparse and potentially irregular meshes. We therefore discuss how to discretise the loss, and propose encoder/decoder architectures that can be evaluated on any mesh.

2.4.1 Encoder Architecture

The encoder f is a map from a function $u\colon \Omega \to \mathbb{R}^m$ to the parameters of the encoder distribution (16a): the mean $f(u;\theta) \in \mathcal{Z}$ and covariance matrix $\Sigma(u;\theta) \in \mathcal{S}_+(\mathcal{Z})$. We assume $\Sigma(u;\theta)$ is diagonal, so f need only return two vectors: the mean $f(u;\theta)$ and the log-diagonal of $\Sigma(u;\theta)$. We thus define

$$\mathsf{f}(u;\theta) = \rho\left(\int_\Omega \kappa\big(x, u(x); \theta\big)\,\mathrm{d}x; \theta\right) \in \mathcal{Z} \times \mathcal{Z} = \mathbb{R}^{2d_\mathcal{Z}}, \tag{25}$$

where $\kappa\colon \Omega \times \mathbb{R}^m \times \Theta \to \mathbb{R}^\ell$ is parametrised as a neural network with two hidden layers of width 64 and output dimension $\ell = 64$, using GELU activation (Hendrycks and Gimpel, 2016), and $\rho\colon \mathbb{R}^\ell \times \Theta \to \mathbb{R}^{2d_\mathcal{Z}}$ is parametrised as a linear layer $\rho(v;\theta) = W^\theta v + b^\theta$, with $W^\theta \in \mathbb{R}^{2d_\mathcal{Z}\times\ell}$ and $b^\theta \in \mathbb{R}^{2d_\mathcal{Z}}$. We augment $x \in \Omega$ with 16 random Fourier features (Appendix B.1) to aid learning of high-frequency features (Tancik et al., 2020). After discretisation on data $\boldsymbol{u} = \{(x_i, u(x_i))\}_{i=1}^I$, in which we approximate the integral over Ω by a normalised sum, our architecture resembles set-to-vector maps such as deep sets (Zaheer et al., 2017), PointNet (Qi et al., 2017), and statistic networks (Edwards and Storkey, 2017), which take the form

$$\big\{\big(x_i, u(x_i)\big) \,\big|\, i = 1, 2, \ldots, I\big\} \mapsto \rho\Big(\mathrm{pool}\Big(\big\{\kappa\big(x_i, u(x_i); \theta\big) \,\big|\, i = 1, 2, \ldots, I\big\}\Big); \theta\Big),$$

where pool is a pooling operation invariant to the order of its inputs—in our case, the mean. Unlike these works we design our architecture for functions and only then discretise; we believe there is great potential to extend other point-cloud and set architectures similarly.

Many other function-to-vector architectures have been proposed, e.g., the variable-input DeepONet (VIDON; Prasthofer et al., 2022), the mesh-independent neural operator (MINO; Lee, 2022) and continuum attention (Calvello et al., 2024), and our proposal is most similar to the linear-functional layer of Fourier neural mappings (Huang et al., 2025) and the neural functional of Rahman et al. (2022). These differ from our approach by preceding (25) by a neural operator; on the problems we consider, we find our encoder map to be equally expressive.

2.4.2 Decoder Architecture

The decoder g is a map from a latent vector $z \in \mathcal{Z}$ to a function $g(z;\psi)\colon \Omega \to \mathbb{R}^m$, which we parametrise using a coordinate neural network $\gamma\colon \mathcal{Z} \times \Omega \times \Psi \to \mathbb{R}^m$ with 5 hidden layers of width 100 using GELU activation throughout, so that

$$g(z;\psi)(x) = \gamma(z, x; \psi). \tag{26}$$

As before, we augment $x \in \Omega$ with 16 random Fourier features (Appendix B.1). Our proposed architecture allows for discretisation of the decoded function $g(z;\psi)$ on any mesh, and the cost of evaluating the decoder (26) grows linearly with the number of mesh points.

There are several related approaches in the literature to parametrise vector-to-function maps. Huang et al. (2025) lift the input by multiplying with a learnable constant function, then apply an operator architecture such as FNO. Seidman et al. (2023) propose both a DeepONet-inspired decoder using a linear combination of learnable basis functions, and a nonlinear decoder essentially the same as what we propose, which is also similar to the

architectures of the nonlinear manifold decoder (NOMAD; Seidman et al., 2022) and PARA-Net (de Hoop et al., 2022). Also related are implicit neural representations (Sitzmann et al., 2020), in which one regresses on a fixed image using a coordinate neural network and treats the resulting weights as a resolution-independent representation of the data.

2.4.3 Discretisation of Per-Sample Loss

To discretise the per-sample losses derived in Section 2.3, we make two approximations. First, we approximate the expectation over $z \sim \mathbb{Q}^{\theta}_{z|u}$ by Monte Carlo sampling (Kingma and Welling, 2014), with the number of samples viewed as a hyperparameter. Second, we approximate the integrals, norms, and inner products arising in the loss, as we now outline.

Per-Sample Loss $\mathcal{L}^{\mathbf{SDE}}_{\lambda,\beta}$. Since the terms appearing in $\mathcal{L}^{\mathrm{SDE}}_{\lambda,\beta}$ are integral functionals of the data and decoded functions, we can discretise on any partition $0 = t_0 < t_1 < \cdots < t_I = T$ and work with data discretised at any time step. The deterministic integral can be approximated by a normalised sum, and the stochastic integral can be discretised as

$$\int_0^T \big\langle g(z;\psi)'(t),\, \mathrm{d}u_t \big\rangle \approx \sum_{i=1}^{I} \Big\langle g(z;\psi)'(t_{i-1}), u(t_i) - u(t_{i-1}) \Big\rangle,$$

which converges in probability in the limit $I \to \infty$ to the true stochastic integral (Särkkä and Solin, 2019, eq. (4.6)). Since the decoder will be a differentiable neural network, terms involving the derivative $g(z;\psi)'(t)$ can be computed using automatic differentiation; we find this to be much more stable than using a finite-difference approximation of the derivative.

Per-Sample Loss $\mathcal{L}^{\mathbf{BIP}}$. For many Bayesian inverse problems, $\mathcal{U}$ is a function space such as $L^2(\Omega)$, and so again the norms and inner products are integral functionals amenable to discretisation on any mesh. However, applying the operator $C^{-1/2}$ is typically tractable only in special cases. One widely used setting is the one in which the eigenbasis of C is known and basis coefficients are readily computable; this arises when C is an inverse power of the Laplacian on a rectangle and a fast Fourier transform may be used.

2.5 Numerical Experiments

We now apply FVAE on two examples where Υ is an SDE path distribution. Both examples serve as prototypes for more complex problems such as those arising in molecular dynamics. For all experiments, we adopt the architecture of Section 2.4. A summary of conclusions to be drawn from the numerical experiments with these examples is as follows:

(a) FVAE captures properties of individual paths as well as ensemble properties of the data set, with the learned latent variables being physically interpretable (Section 2.5.1);

(b) choosing decoder noise that accurately reflects the stochastic variability in the data is essential to obtain a high-quality generative model (Section 2.5.1);

(c) FVAE is robust to changes of mesh in the encoder and decoder, enabling training with heterogeneous data and generative modelling at any resolution (Section 2.5.2).

We emphasise in these experiments that FVAE does this purely from data, with no knowledge of the data-generating process other than in the choice of decoder noise.

2.5.1 Brownian Dynamics

The Brownian dynamics model (see Schlick, 2010, Chap. 14), also known as the Langevin model, is a stochastic approximation of deterministic Newtonian models for molecular dynamics. In this model, the configuration u_t (in some configuration space $X \subseteq \mathbb{R}^m$) of a molecule is assumed to follow the gradient flow of a **potential** $U\colon X \to \mathbb{R}$ perturbed by additive thermal noise with **temperature** $\varepsilon > 0$. This leads to the Langevin SDE

$$\mathrm{d}u_t = -\nabla U(u_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \qquad t \in [0, T], \tag{27}$$

where $(w_t)_{t\in[0,T]}$ is a Brownian motion on $\mathbb{R}^m$. As a prototype for the more sophisticated, high-dimensional potentials arising in molecular dynamics, such as the Lennard–Jones potential (Schlick, 2010), we take $X = \mathbb{R}$ and consider the asymmetric double-well potential

$$U(x) \propto 3x^4 + 2x^3 - 6x^2 - 6x. \tag{28}$$

This has a local minimum at $x_1 = -1$ and a global minimum at $x_2 = +1$ (Figure 1(a)). We take Υ to be the corresponding path distribution, with temperature $\varepsilon = 1$, final time $T = 5$, and initial condition $u_0 = x_1$. (The preceding developments fixed $u_0 = 0$ but are readily adapted to any fixed initial condition.) The training data set consists of 8,192 paths with time step $5/512$ in $[0, T]$, and in each path 50% of time steps are missing (see Appendix B.2).

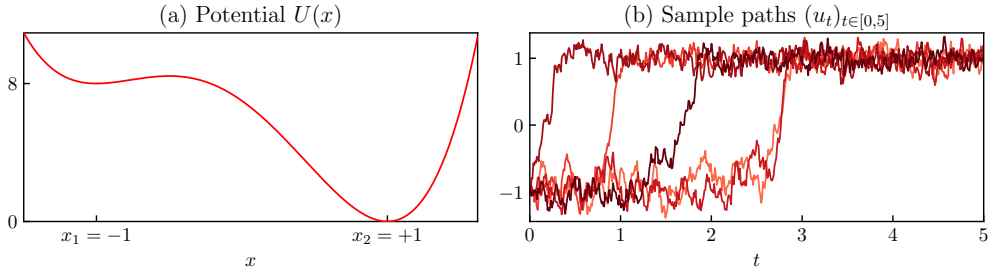


Figure 1: (a) Realisations of the SDE (27) follow the gradient flow of the potential U. (b) Sample paths $(u_t)_{t\in[0,T]} \sim \Upsilon$ begin at $x_1 = -1$ and transition with high probability to the lower-potential state $x_2 = +1$ as a result of the additive thermal noise.

Sample paths drawn from Υ start at x_1 and transition with very high probability to the potential-minimising state x_2; the time at which the transition begins is determined by the thermal noise, but, once the transition has begun, the manner in which the transition occurs is largely consistent across realisations. Such universal transition phenomena occur quite generally in the study of random dynamical systems, as a consequence of large-deviation theory (see E et al., 2004).

We train FVAE using the SDE loss (Section 2.3.1) with regularisation parameter $\beta = 1.2$ and zero-penalty scale $\lambda = 10$. Motivated by the observation that trajectories are determined chiefly by the transition time, we use latent dimension $d_{\mathcal{Z}} = 1$.

Choice of Noise Process. The choice of decoder noise greatly affects FVAE's performance as an autoencoder and a generative model. To investigate this, we first train three instances of FVAE with different restoring forces κ in the decoder-noise process. Then, to evaluate autoencoding performance, we draw samples $u \sim \Upsilon$ from the held-out set and compute

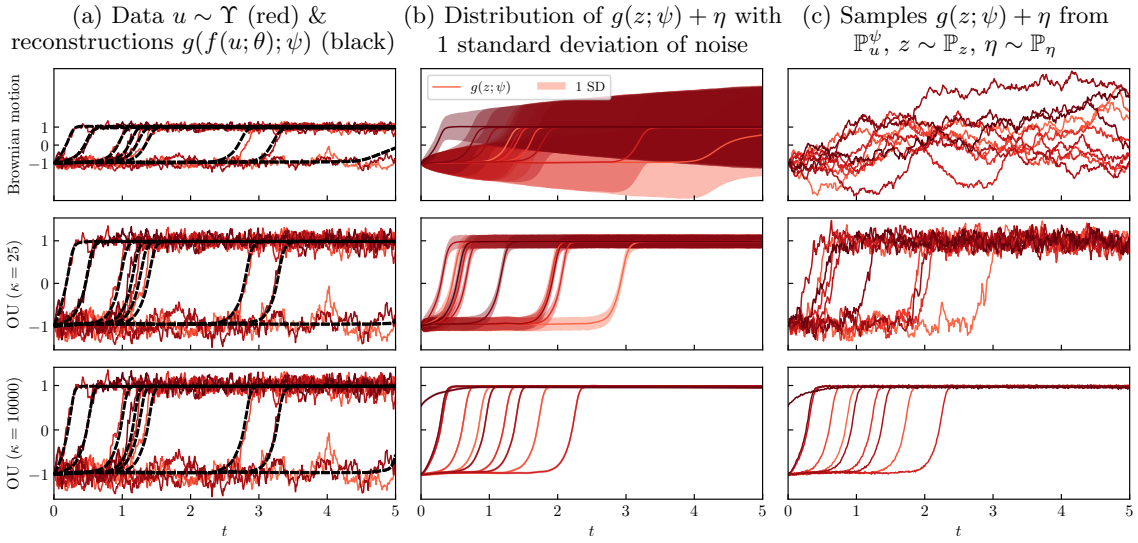


Figure 2: The SDE loss gives much freedom in the choice of noise process $(\eta_t)_t$. The top row uses Brownian motion as the decoder noise and the second and third rows use OU processes with different asymptotic variances. While all choices lead to high-quality reconstructions, only the OU process with $\kappa = 25$ gives a generative model that agrees well with the data.

the *reconstruction* $g(f(u;\theta);\psi)$, which is the mean of the decoder distribution $\mathbb{P}^{\psi}_{u|z}$ with $z = f(u;\theta)$ taken to be the mean of the encoder distribution $\mathbb{Q}^{\theta}_{z|u}$ (Figure 2(a)). To evaluate FVAE as a generative model, we draw samples from the latent distribution $\mathbb{P}_z$ and display the mean $g(z;\psi)$ of the decoder distribution $\mathbb{P}^{\psi}_{u|z}$ along with a shaded region indicating one standard deviation of the noise process (Figure 2(b)); moreover we draw samples $g(z;\psi)+\eta$ to illustrate their qualitative behaviour (Figure 2(c)).

Using Brownian motion as the decoder noise leads to excellent reconstructions, but samples from the generative model appear different from the training data. By using OU-distributed noise with restoring force $\kappa > 0$ we obtain similar reconstructions to those achieved under Brownian motion, but samples from the generative model match the data distribution more closely. This is because the variance of Brownian motion grows unboundedly with time, while the asymptotic variance under the OU process is $\varepsilon/2\kappa$, better reflecting the behaviour of the data for well-chosen κ.

On this data set, choosing a suitable noise process $(\eta_t)_t$ has the added benefit of significantly accelerating training: autoencoding mean-squared error (MSE) decreases much faster under OU noise ($\kappa > 0$) than under Brownian motion noise (Figure 3). We expect the choice of noise should depend in general on properties of the data distribution, with the OU process being particularly suited to this data set; in the discussion that follows, we use an OU process with restoring force $\kappa = 25$.

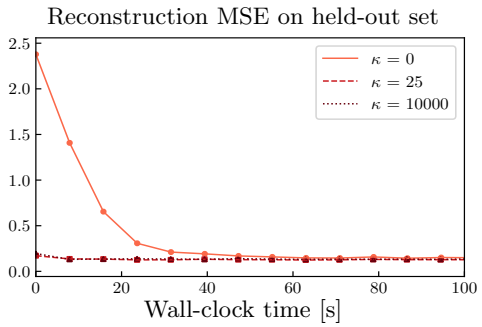


Figure 3: Using OU noise ($\kappa > 0$) leads to faster training convergence.

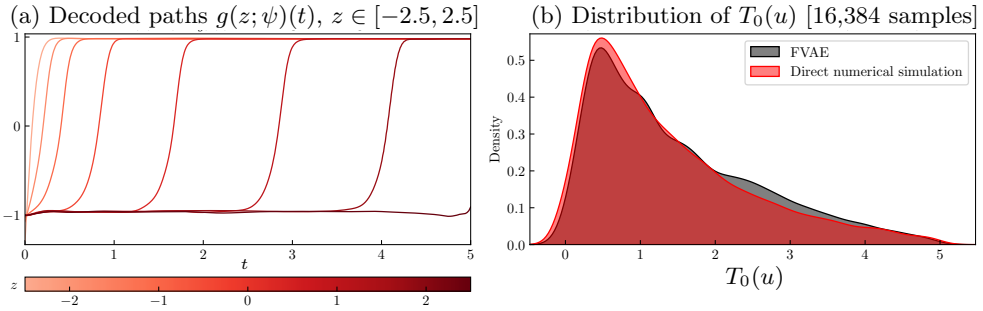


Figure 4: (a) The latent variable z identified by FVAE corresponds to the first-crossing time T_0 of the decoded path $g(z;\psi)$. (b) Kernel density estimates of the distributions of T_0 under the FVAE generative model, and when computed using direct simulations, closely agree.

Unsupervised Learning of Physically Relevant Quantities. Our choice of latent dimension $d_{\mathcal{Z}} = 1$ was motivated by the heuristic that the time of the transition from $x_1 = -1$ to $x_2 = +1$ essentially determines the SDE trajectory. FVAE identifies this purely from data, with the learned latent variable $z \in \mathcal{Z}$ being in correspondence with the transition time: larger values of z map to paths $g(z;\psi)$ transitioning later in time (Figure 4(a)).

To understand whether FVAE captures ensemble statistical properties of the data distribution, we compare the distributions of the first-crossing time $T_0(u) = \inf\{t > 0 \,|\, u_t \geqslant 0\}$ estimated using 16,384 paths from the generative model and 16,384 direct simulations, using kernel density estimates based on Gaussian kernels with bandwidths selected by Scott's rule (Scott, 2015, eq. (6.44)); we find that the two distributions closely agree (Figure 4(b)).

2.5.2 Estimation of Markov State Models

In practical applications of molecular dynamics, one is often interested in the evolution of large molecules on long timescales. For example in the study of protein folding (Konovalov et al., 2021), it is of interest to capture the complex, multistage transitions of proteins between configurations. Moving beyond the toy one-dimensional problem in Section 2.5.1, the chief difficulty is the very high dimension of such systems, which makes simulations possible only on timescales orders of magnitudes shorter than those of physical interest.

Markov state models (MSMs) offer one method of distilling many simulations on short timescales into a statistical model permitting much longer simulations (Husic and Pande, 2018). Assuming that the dynamics are given by a random process $(u_t)_{t\geqslant 0}$ taking values in the configuration space X, an MSM can be constructed by partitioning X into disjoint state sets $X = X_1 \cup \cdots \cup X_p$, and, for some **lag time** $\tau > 0$, considering the discrete-time process $(U_k)_{k\in\mathbb{N}}$ for which $U_k = i$ if and only if $u_{k\tau} \in X_i$. One hopes that, if τ is sufficiently large, the process $(U_k)_{k\in\mathbb{N}}$ is approximately Markov, and thus its distribution can be characterised by learning the probabilities of transitioning in time τ from one state to another. These probabilities can be determined using the short-run simulations—which can be generated in parallel—and the resulting MSM can be used to simulate on longer timescales.

Motivated by this application, we consider the problem of constructing an MSM from data provided at sparse or irregular intervals that do not necessarily align with the lag time τ; in this case, computing the probability of transition in time τ directly may not be possible.

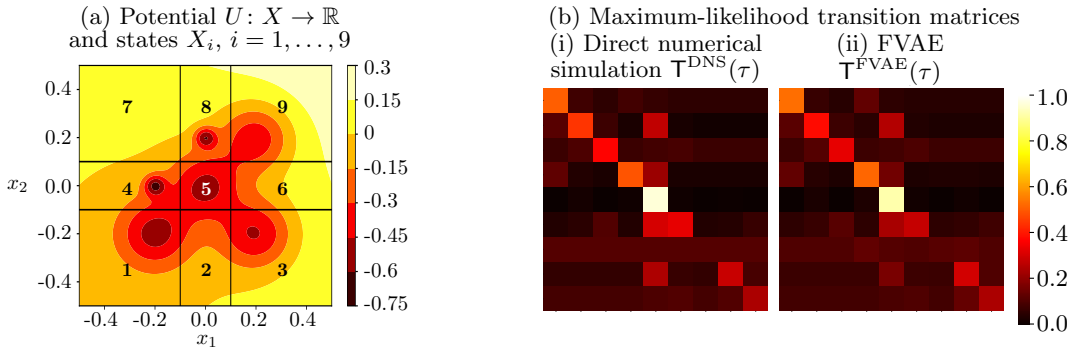


Figure 5: (a) Contour plot of the potential U and the division of the state space X. All paths start at $t = 0$ at the origin (in state 5). (b) Transition matrices with lag $\tau = 3/512$ computed using FVAE and through direct simulation, both on the time interval $[0, 3]$.

We show the power of FVAE in this problem by first learning a generative model from the heterogeneously sampled data and then using the generative model to draw paths sampled at the regular time step τ; constructing an MSM from these paths is then straightforward.

We give an example based on the Brownian dynamics model (27) on $X = \mathbb{R}^2$ using a multiwell potential U (Figure 5(a)), stated precisely in Appendix B.3, which we take to be a quadratic bowl, perturbed by a linear function to break the symmetry, and by six Gaussian densities to act as potential wells with minima at $(0, 0)$, $(0.2, 0.2)$, $(-0.2, -0.2)$, $(0.2, -0.2)$, $(0, 0.2)$ and $(-0.2, 0)$. We take $\mathcal{U} = C_0([0, T], X)$ and let $\Upsilon \in \mathscr{P}(\mathcal{U})$ be the path distribution, with temperature $\varepsilon = 0.1$, final time $T = 3$ and initial condition $u_0 = 0$. The training data set consists of 16,384 paths discretised with time step $3/512$, where, for each sample, it is assumed that 50% of steps are missing (details in Appendix B.3). We train FVAE using the SDE loss (Section 2.3.1) with $\kappa = 100$, $\lambda = 50$, $\beta = 0.02$, and latent dimension $d_{\mathcal{Z}} = 16$.

Partitioning the Configuration Space. The states of an MSM can be selected manually using expert knowledge or automatically using variational or machine-learning methods (Mardt et al., 2018). For simplicity, we choose the states by hand, partitioning $X = \mathbb{R}^2$ into $p = 9$ disjoint regions (Figure 5(a)) divided by the four lines $x_1 = \pm 0.1$ and $x_2 = \pm 0.1$.

Estimating Transition Probabilities with FVAE. After training FVAE with irregularly sampled data, we draw samples from the generative model with regular time step τ and use these samples to compute the MSM transition probabilities. Setting aside the question of Markovianity for simplicity, we draw from the generative model $M = 2{,}048$ paths $\{v^{(m)}\}_{m=1}^{M}$ discretised on a mesh of $K = 513$ equally spaced points with time step $\tau = 3/512$ on $[0, T]$, and compute the count matrix

$$\mathsf{C}^{\mathrm{FVAE}}(\tau) = \left(\mathsf{C}^{\mathrm{FVAE}}_{ij}(\tau)\right)_{i,j\in\{1,\ldots,p\}}, \quad \mathsf{C}^{\mathrm{FVAE}}_{ij}(\tau) = \sum_{k=0}^{K}\sum_{m=1}^{M} \mathbb{1}\left[v^{(m)}_{k\tau} \in X_i \text{ and } v^{(m)}_{(k+1)\tau} \in X_j\right].$$

We then derive the corresponding maximum-likelihood transition matrix $\mathsf{T}^{\mathrm{FVAE}}(\tau)$ by normalising each row of $\mathsf{C}^{\mathrm{FVAE}}(\tau)$ to sum to one; for simplicity we do not constrain the transition matrix to satisfy the detailed-balance condition (see Prinz et al., 2011, Sec. IV.D).

The resulting transition matrix $\mathsf{T}^{\mathrm{FVAE}}(\tau)$ agrees closely with the matrix $\mathsf{T}^{\mathrm{DNS}}(\tau)$ computed analogously using 2,048 direct numerical simulations on the regular time step τ (Figure 5(b)).

3. Problems with VAEs in Infinite Dimensions

As we have seen in Section 2.1, the empirical FVAE objective $\mathcal{J}_N^{\mathrm{FVAE}}(\theta,\psi)$ for the data set $\{u^{(n)}\}_{n=1}^N$ is based on a sequence of approximations and equalities:

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(u^{(n)};\theta,\psi)}_{=:\mathcal{J}_N^{\mathrm{FVAE}}(\theta,\psi)} \underset{\text{(A)}}{\approx} \underbrace{\mathbb{E}_{u\sim\Upsilon}[\mathcal{L}(u;\theta,\psi)]}_{=:\mathcal{J}^{\mathrm{FVAE}}(\theta,\psi)} \underset{\text{(B)}}{=} D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u}\,\|\,\mathbb{P}^\psi_{z,u}) - \underbrace{D_{\mathrm{KL}}(\Upsilon\,\|\,\Lambda)}_{\text{“constant”}}.$$

The approximation (A) is based on the law of large numbers and is an equality almost surely in the limit $N\to\infty$. The equality (B) is true by Theorem 6 with a finite constant $D_{\mathrm{KL}}(\Upsilon\,\|\,\Lambda)$ provided Assumption 5 holds—but if the assumption does not hold, this “constant” may well be infinite. So, while it is tempting to apply the empirical objective $\mathcal{J}_N^{\mathrm{FVAE}}$ without first checking the validity of (A) and (B), this strategy is fraught with pitfalls.

To illustrate this we apply FVAE in the white-noise setting of Example 1; this coincides with the setting of the VANO model (Seidman et al., 2023), which we discuss in detail in the related work (Section 5). In this example we derive the per-sample loss $\mathcal{L}$ and apply the resulting empirical objective $\mathcal{J}_N^{\mathrm{FVAE}}$ for training; but both the joint divergence $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u}\,\|\,\mathbb{P}^\psi_{z,u})$ and the constant $D_{\mathrm{KL}}(\Upsilon\,\|\,\Lambda)$ turn out to be infinite. Consequently, the approximations (A) and (B) break down, which we see numerically: discretisations of $\mathcal{J}_N^{\mathrm{FVAE}}$ appear to diverge as resolution is refined, suggesting that they have no continuum limit.

Example 2 *Take $\mathcal{U}=H^{-1}([0,1])$ and assume that Υ is the distribution of u in the model*

$$\begin{aligned}\xi &\sim \mathrm{Uniform}[0,1]\\ u\mid\xi &= \delta_\xi.\end{aligned}$$

Realisations of u in this model lie in $H^s([0,1])$, $s<-1/2$, with probability one, so $\Upsilon\in\mathscr{P}(\mathcal{U})$. This data can be viewed as a prototype for rough behaviour such as the derivative of shock profiles arising in hyperbolic PDEs with random initial data. It will serve as an extreme example allowing us to isolate the numerical issues associated with using FVAE or VANO in the misspecified setting.

Take the model (16a)–(16d) *for real-valued functions on $[0,1]$, with $\mathbb{P}_\eta$ taken to be L^2-white noise. As discussed in Proposition 15, $\mathbb{P}_\eta\in\mathscr{P}(H^s([0,1]))$, $s<-1/2$, and $H(\mathbb{P}_\eta)=L^2([0,1])$. Fixing the reference distribution $\Lambda=\mathbb{P}_\eta$ and assuming g takes values in $L^2([0,1])$, the Cameron–Martin theorem (Proposition 19) ensures that the density $\mathrm{d}\mathbb{P}^\psi_{u|z}/\mathrm{d}\Lambda$ exists, and consequently*

$$\mathcal{L}(u;\theta,\psi)=\mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\Big[\tfrac{1}{2}\|g(z;\psi)\|_{L^2}^2-\langle g(z;\psi),u\rangle_{L^2}^{\sim}\Big]+D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u}\,\big\|\,\mathbb{P}_z\big). \tag{29}$$

At this stage, we have not verified that the joint divergence $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z,u}\,\|\,\mathbb{P}^\psi_{z,u})$ has finite infimum, nor that Assumption 5 holds; indeed we will see that both of these conditions fail.

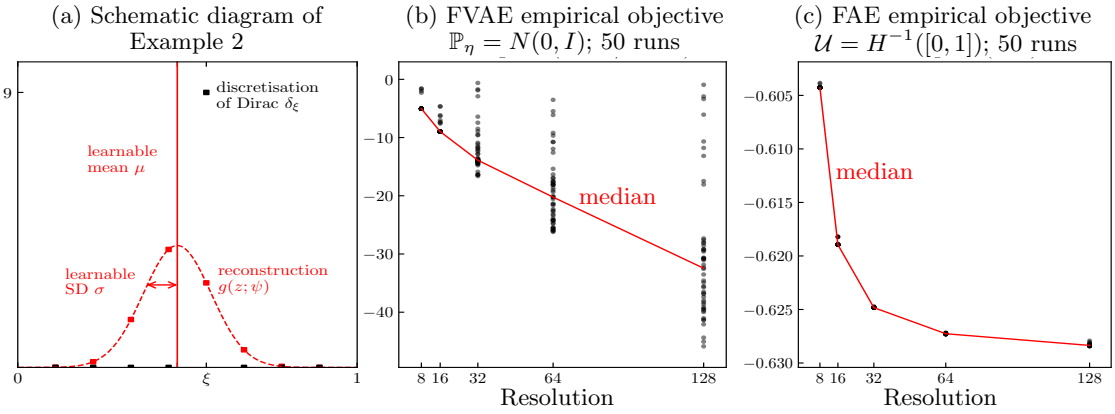


Figure 6: (a) The discrete representations (squares) of δ_ξ and $g(z;\psi)$ on a grid of 8 points. (b) In Example 2, the FVAE empirical objective at the minimising parameters diverges as resolution is increased. (c) To overcome this, we propose a regularised autoencoder, FAE, in Section 4. Repeating the experiment with this objective suggests that the FAE empirical objective has a well-defined continuum limit.

Nevertheless we can attempt to train FVAE with the empirical objective resulting from (29)*. To do this we choose* $\mathcal{Z} = \mathbb{R}$ *and use an encoder map* f *and a decoder map* g *tailored to this problem, since the architectures of Section 2.4 are not equipped to deal with functions of negative Sobolev regularity. We parametrise* f *as*

$$\mathsf{f}(u;\theta) = \rho\left(\arg\max_{x\in[0,1]}(\varphi * u)(x); \theta\right) \in \mathcal{Z}\times\mathcal{Z} = \mathbb{R}^2,$$

where ρ *is a neural network and* φ *is a compactly supported smooth mollifier, chosen such that* $\varphi * u$ *has well-defined maximum, and we parametrise* g *to return the Gaussian density*

$$g\big(z;\psi\big)(x) = N\big(\mu(z;\psi), \sigma(z;\psi)^2; x\big),$$

with mean $\mu(z;\psi) \in [0,1]$ *and standard deviation* $\sigma(z;\psi) > 0$ *computed from a neural network (see Appendix B.4). In other words,* f *applies a neural network to the location of the maximum of* u*, while* g *returns a Gaussian density with learned mean and variance (Figure 6(a)). To investigate the behaviour of discretisations of* $\mathcal{J}_N^{FVAE}$ *as resolution is refined, we generate a sequence of data sets in which we discretise on a mesh of* $I \in \{8, 16, 32, 64, 128\}$ *equally spaced points* $\{i/I+1\}_{i=1,\dots,I} \subset [0,1]$*. At each resolution we generate* I *training samples: one discretised Dirac function at each mesh point, normalised to have unit* L^1*-norm. We train 50 independent instances of FVAE at each resolution and record the value of the empirical objective at convergence (Figure 6(a)). Notably, the empirical objective appears to diverge as resolution is refined. This has two major causes:*

(a) Υ *is not absolutely continuous with respect to* $\Lambda = \mathbb{P}_\eta$*: the set* $\{\delta_x \mid x \in [0,1]\}$ *has probability zero under* Λ *but probability one under* Υ*; thus* $D_{\mathrm{KL}}(\Upsilon \,\|\, \Lambda) = \infty$*.*

(b) Υ *is not absolutely continuous with respect to* $\mathbb{P}_u^\psi$*, meaning that* $D_{\mathrm{KL}}(\mathbb{Q}_{z,u}^\theta \,\|\, \mathbb{P}_{z,u}^\psi) = \infty$ *for all* θ *and* ψ*. To see this, we again note that* $\{\delta_x \mid x \in [0,1]\}$ *has probability one*

under Υ, but, as a consequence of the Cameron–Martin theorem, $\mathbb{P}^{\psi}_{u|z}$ and $\mathbb{P}_\eta$ are mutually absolutely continuous, and thus as $\mathbb{P}_\eta(\{\delta_x \mid x \in [0,1]\}) = 0$,

$$\mathbb{P}^{\psi}_{u}\big(\{\delta_x \mid x \in [0,1]\}\big) = \int_{\mathcal{Z}} \mathbb{P}^{\psi}_{u|z}\big(\{\delta_x \mid x \in [0,1]\}\big)\, \mathbb{P}_z(\mathrm{d}z) = 0.$$

The problematic term in the per-sample loss is the measurable linear functional $\langle g(z;\psi), u\rangle^{\sim}_{L^2}$; this is defined only up to modification on $\mathbb{P}_\eta$-probability zero sets, and yet we evaluate it on just such sets—namely, the $\mathbb{P}_\eta$-probability zero set $\{\delta_x \mid x \in [0,1]\}$. ■

Remark 22 *The joint divergence $D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z,u} \,\|\, \mathbb{P}^{\psi}_{z,u})$ would also be infinite if Υ was supported on $L^2([0,1])$, as seen in Example 1, but it is harder to observe any numerical issue in training. This is because the measurable linear functional $\langle g(z;\psi), u\rangle^{\sim}_{L^2}$ reduces to the usual L^2-inner product, and so, even though the FVAE objective is not well defined, the per-sample loss can still be viewed as a regularised misfit (see Theorem 20):*

$$\mathcal{L}(u;\theta,\psi) = \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^{\theta}_{z|u}}\left[\tfrac{1}{2}\|g(z;\psi) - u\|^2_{L^2}\right] + D_{\mathrm{KL}}(\mathbb{Q}^{\theta}_{z|u} \,\|\, \mathbb{P}_z) - \tfrac{1}{2}\|u\|^2_{L^2}.$$

As a result one can reinterpret the objective as that of a regularised autoencoder (see Remark 11). This motivates our use of a regularised autoencoder, FAE, in Section 4. ■

4. Regularised Autoencoders on Function Space

To overcome the issues in applying VAEs in infinite dimensions, we set aside the probabilistic motivation for FVAE and define a regularised autoencoder in function space, the **functional autoencoder (FAE)**, avoiding the need for onerous conditions on the data distribution. In Section 4.1, we state the FAE objective and make connections to the FVAE objective. Section 4.2 outlines minor adaptations to the FVAE encoder and decoder for use with FAE. Section 4.3 demonstrates FAE on two examples from the sciences: incompressible fluid flows governed by the Navier–Stokes equation; and porous-medium flows governed by Darcy's law.

4.1 Training Objective

Throughout Section 4 we make the following assumption on the data, postponing discussion of discretisation to Section 4.2. Unlike in Section 2 we do not need $\mathcal{U}$ to be separable, allowing us to consider data from an even wider variety of spaces, such as the (non-separable) space $\mathrm{BV}(\Omega)$ of bounded-variation functions on $\Omega \subset \mathbb{R}^d$.

Assumption 23 *Let $(\mathcal{U}, \|\cdot\|)$ be a Banach space. There exists a data distribution $\Upsilon \in \mathscr{P}(\mathcal{U})$ from which we have access to N independent and identically distributed samples $\{u^{(n)}\}_{n=1}^N \subset \mathcal{U}$.* ■

To define our regularised autoencoder, we fix a latent space $\mathcal{Z} = \mathbb{R}^{d_Z}$ and define encoder and decoder transformations f and g, which, unlike in FVAE, return points rather than probability distributions:

$$\text{(encoder)} \qquad \mathcal{U} \ni u \mapsto f(u;\theta) \in \mathcal{Z}, \tag{30a}$$

$$\text{(decoder)} \qquad \mathcal{Z} \ni z \mapsto g(z;\psi) \in \mathcal{U}. \tag{30b}$$

We then take as our objective the sum of a misfit term between the data and its reconstruction, and a regularisation term with **regularisation parameter** $\beta > 0$ on the encoded vectors:

$$\text{(FAE objective)} \qquad \mathcal{J}_\beta^{\mathrm{FAE}}(\theta, \psi) = \mathop{\mathbb{E}}_{u\sim\Upsilon}\Big[\tfrac{1}{2}\big\|g\big(f(u;\theta);\psi\big) - u\big\|^2 + \beta\big\|f(u;\theta)\big\|_2^2\Big]. \tag{31}$$

As in Section 2.1, the expectation over $u \sim \Upsilon$ is approximated by an average over the training data, resulting in the empirical objective $\mathcal{J}_{\beta,N}^{\mathrm{FAE}}$. There is great flexibility in the choice of regularisation term; we adopt the squared Euclidean norm $\|f(u;\theta)\|_2^2$ as a simplifying choice consistent with using a Gaussian prior $\mathbb{P}_z$ in a VAE (Remark 11). While (31) has much in common with the FVAE objective $\mathcal{J}^{\mathrm{FVAE}}$, it is not marred by the foundational issues raised in Section 3; indeed, the FAE objective is broadly applicable as the following result shows.

Proposition 24 *Suppose Υ has finite second moment, i.e., $\mathbb{E}_{u\sim\Upsilon}\big[\|u\|^2\big] < \infty$. If there exist $\theta^\star \in \Theta$ and $\psi^\star \in \Psi$ such that $f(u;\theta^\star) = 0$ and $g(z;\psi^\star) = 0$, then (31) has finite infimum.*

Proof This follows immediately from evaluating $\mathcal{J}_\beta^{\mathrm{FAE}}$ at $\theta^\star$ and $\psi^\star$, since $\mathcal{J}_\beta^{\mathrm{FAE}}(\theta^\star, \psi^\star) = \mathbb{E}_{u\sim\Upsilon}\big[\frac{1}{2}\|u\|^2\big]$, and the expectation is finite by hypothesis. ∎

4.2 Architecture and Algorithms

To train FAE we must discretise the objective $\mathcal{J}_\beta^{\mathrm{FAE}}$ and parametrise the encoder f and decoder g with learnable maps. We moreover propose a masked training scheme that appears to be new to the operator-learning literature; as we will show in Section 4.3, this scheme greatly improves the robustness of FAE to changes of mesh.

Encoder and Decoder Architecture. As in Section 2.4, we will construct encoder and decoder architectures under the assumption that $\mathcal{U}$ is a Banach space of functions evaluable pointwise almost everywhere with domain $\Omega \subseteq \mathbb{R}^d$ and range $\mathbb{R}^m$, and that we have access to discretisations $\boldsymbol{u}^{(n)}$ of the data $u^{(n)}$ comprised of evaluations at finitely many mesh points. We adopt an architecture near identical to that used for FVAE. More precisely, we parametrise the encoder as

$$f(u;\theta) = \rho\left(\int_\Omega \kappa\big(x, u(x);\theta\big)\,\mathrm{d}x;\theta\right) \in \mathcal{Z},$$

with $\kappa\colon \Omega \times \mathbb{R}^m \times \Theta \to \mathbb{R}^\ell$ parametrised as a neural network with two hidden layers of width 64, output dimension $\ell = 64$, and with $\rho\colon \mathbb{R}^\ell \times \Theta \to \mathbb{R}^{d_z}$ parametrised as the linear layer $\rho(v;\theta) = W^\theta v + b^\theta$. We parametrise the decoder as the coordinate neural network $\gamma\colon \mathcal{Z} \times \Omega \times \Psi \to \mathbb{R}^m$ with 5 hidden layers of width 100, so that

$$g(z;\psi)(x) = \gamma(z, x;\psi) \in \mathbb{R}^m.$$

In both cases we use GELU activation and augment x with 16 random Fourier features (Appendix B.1). Relative to the architectures of Section 2.4, the only change is in the range of f, which now takes values in $\mathcal{Z}$ instead of returning distributional parameters for $\mathbb{Q}_{z|u}^\theta$.

Discretisation of $\mathcal{J}_\beta^{\mathbf{FAE}}$. The FAE objective can be applied whenever $\mathcal{U}$ is Banach, but in this article we take $\mathcal{U} = L^2(\Omega)$, where $\Omega = \mathbb{T}^d$ or $\Omega = [0,1]^d$, and discretise the $\mathcal{U}$-norm with a normalised sum. One can readily imagine other possibilities, e.g., taking $\mathcal{U}$ to be a Sobolev space of order $s \geqslant 1$ if derivative information is available (Czarnecki et al., 2017), and approximating the L^2-norm of the data and its derivatives by sums. More generally we may use linear functionals as the starting point for approximation.

4.2.1 Masked Training

Self-supervised training—learning to predict the missing data from masked inputs—has proven valuable in both language models such as BERT (Devlin et al., 2019) and vision models such as the masked autoencoder (MAE; He et al., 2022). This method has been shown to both reduce training time and to improve generalisation. We propose two schemes making use of the ability to discretise the encoder and decoder on arbitrary meshes: **complement masking** and **random masking**. Under both schemes we transform each discretised training sample $\boldsymbol{u} = \{(x_i, u(x_i))\}_{i=1}^I$ by subsampling with index sets $\mathcal{I}_{\text{enc}}$ and $\mathcal{I}_{\text{dec}}$, which may change at each training step, to obtain

$$\boldsymbol{u}_{\text{enc}} = \left\{ \big(x_i, u(x_i)\big) \,\middle|\, i \in \mathcal{I}_{\text{enc}} \right\}, \qquad \boldsymbol{u}_{\text{dec}} = \left\{ \big(x_i, u(x_i)\big) \,\middle|\, i \in \mathcal{I}_{\text{dec}} \right\}.$$

We supply $\boldsymbol{u}_{\text{enc}}$ as input to the discretised encoder; moreover we discretise the decoder on the mesh $\{x_i\}_{i \in \mathcal{I}_{\text{dec}}}$ and compare the decoder output to the masked data $\boldsymbol{u}_{\text{dec}}$. In both of the strategies we propose $\mathcal{I}_{\text{enc}}$ and $\mathcal{I}_{\text{dec}}$ will be unstructured random subsets of $\{1, \dots, I\}$, but in principle other masks—such as polygons—could be considered.

Complement Masking. The chief strategy used in the numerical experiments is to draw $\mathcal{I}_{\text{enc}}$ as a random subset of $\{1, \dots, I\}$ and take $\mathcal{I}_{\text{dec}} = \{1, \dots, I\} \setminus \mathcal{I}_{\text{enc}}$, fixing the **(encoder) point ratio** $r_{\text{enc}} = |\mathcal{I}_{\text{enc}}|/I$ as a hyperparameter. This is similar to the strategy adopted by MAE—though our approach differs by masking individual mesh points instead of patches. We explore the tradeoffs in the choice of point ratio in Section 4.3.1.

Random Masking. A second strategy we consider is to independently draw $\mathcal{I}_{\text{enc}}$ and $\mathcal{I}_{\text{dec}}$ as random subsets of $\{1, \dots, I\}$, fixing both the encoder point ratio $r_{\text{enc}} = |\mathcal{I}_{\text{enc}}|/I$ and the decoder point ratio $r_{\text{dec}} = |\mathcal{I}_{\text{dec}}|/I$. This gives greater control of the cost of evaluating the encoder and decoder: by taking both r_{enc} and r_{dec} to be small, we significantly reduce the cost of each training step, which is useful when the number of mesh points I is large.

4.3 Numerical Experiments

In Sections 4.3.1 and 4.3.2, we apply FAE as an out-of-the-box method to discover a low-dimensional latent space for solutions to the incompressible Navier–Stokes equations and the Darcy model for flow in a porous medium, respectively. We find that for these data sets:

(a) FAE's mesh-invariant architecture autoencodes with performance comparable to convolutional neural network (CNN) architectures of similar size (Section 4.3.1);

(b) the ability to discretise the encoder and decoder on different meshes enables new applications to inpainting and data-driven superresolution (Section 4.3.1), as well as extensions of existing zero-shot superresolution as proposed for VANO by Seidman et al. (2023);

(c) masked training significantly improves performance under mesh changes (Section 4.3.1) and can accelerate training while reducing memory demand (Section 4.3.2);

(d) training a generative model on the FAE latent space leads to a resolution-invariant generative model which accurately captures distributional properties (Section 4.3.2).

4.3.1 Incompressible Navier–Stokes Equations

We first illustrate how FAE can be used to learn a low-dimensional representation for snapshots of the vorticity of a fluid flow in two spatial dimensions governed by the incompressible Navier–Stokes equations, and illustrate some of the benefits of our mesh-invariant model.

Let $\Omega = \mathbb{T}^2$ be the torus, viewed as the square $[0,1]^2$ with opposing edges identified and with unit normal $\hat{z}$, and let $\mathcal{U} = L^2(\Omega)$. While the incompressible Navier–Stokes equations are typically formulated in terms of the primitive variables of velocity u and pressure p, it is more natural in this case to work with the vorticity–streamfunction formulation (see Chandler and Kerswell, 2013, eq. (2.6)). In particular the vorticity $\nabla \times u$ is zero except in the out-of-plane component $\hat{z}\omega$. The scalar component of the vorticity, ω, then satisfies

$$\begin{aligned} \partial_t \omega &= \hat{z} \cdot \big(\nabla \times (u \times \omega \hat{z})\big) + \nu \Delta \omega + \varphi, && (x,t) \in \Omega \times (0,T], \\ \omega(x,0) &= \omega_0(x), && x \in \Omega. \end{aligned} \tag{32}$$

In this setting the velocity is given by $u = \nabla \times (\psi_s \hat{z})$, where the streamfunction ψ_s satisfies $\omega = \Delta \psi_s$.[1] Thus, using periodicity, ψ_s is uniquely defined, up to an irrelevant constant, in terms of ω, and (32) defines a closed evolution equation for ω. We suppose that $\Upsilon \in \mathscr{P}(\mathcal{U})$ is the distribution of the scalar vorticity $\omega(\,\cdot\,, T = 50)$ given by (32), with viscosity $\nu = 10^{-4}$ and forcing

$$\varphi(x) = \tfrac{1}{10}\sin(2\pi x_1 + 2\pi x_2) + \tfrac{1}{10}\cos(2\pi x_1 + 2\pi x_2).$$

We assume that the initial condition ω_0 has distribution $N(0, C)$ with covariance operator $C = 7^{3/2}(49I - \Delta)^{-5/2}$, where Δ is the Laplacian operator for spatially-mean-zero functions on the torus $\mathbb{T}^2$. The training data set, based on that of Li et al. (2021), consists of 8,000 samples from Υ generated on a 64×64 grid using a pseudospectral solver, with a further 2,000 independent samples held out as an evaluation set. The data are scaled so that $\omega(x,T) \in [0,1]$ for all $x \in \Omega$. Further details are provided in Appendix B.5.

We train FAE with latent dimension $d_{\mathcal{Z}} = 64$ and regularisation parameter $\beta = 10^{-3}$, and train with complement masking using a point ratio r_{enc} of 30%.

Performance at Fixed Resolution. We compare the autoencoding performance of our mesh-invariant FAE architecture to a standard fixed-resolution CNN architecture, both trained using the FAE objective with all other hyperparameters the same. Our goal is to understand whether our architecture is competitive even without the inductive bias of CNNs.

To do this we fix a class of FAE and CNN architectures for 64×64 unmasked data, and perform a search to select the best-performing models with similar parameter counts (details in Appendix B.5). The FAE architecture achieves reconstruction MSE slightly greater than the CNN on the held-out data, with a comparable number of parameters (Table 1). It is reasonable to expect that mesh-invariance of the FAE architecture comes at some cost to

1. While it is typical in the literature (e.g., Chandler and Kerswell, 2013) to denote the streamfunction by ψ, we instead denote it by ψ_s to avoid ambiguity with the decoder parameter ψ.

Autoencoding on evaluation set (64×64 grid)		
	MSE	Parameters
FAE architecture	4.82×10^{-4} $\pm 2.57 \times 10^{-5}$	64,857
CNN architecture	2.38×10^{-4} $\pm 9.43 \times 10^{-6}$	71,553
Mean $\pm$ 1 standard deviation; 5 training runs		

Table 1: Our resolution-invariant architecture performs comparably to CNNs with similar parameter counts.

performance at fixed resolution, especially as the CNN benefits from a strong inductive bias, but the results of Table 1 suggest that the cost is modest. Further research is desirable to close this gap through better mesh-invariant architectures.

Inpainting. Methods for inpainting—inferring missing parts of an input based on observations and prior knowledge from training data—and related inverse problems are of great interest in computer vision and in scientific applications (Quan et al., 2024). We exploit the ability of FAE to encode on any mesh, and decode on any other mesh, to solve a variety of inpainting tasks. More precisely, after training FAE, we take data from the held-out set on a 64×64 grid and, for each discretised sample, we apply one of three possible masks:

(i) random masking with point ratio 5%, i.e., masking 95% of mesh points; or

(ii) masking of all mesh points lying in a square with random centre and side length; or

(iii) masking of all mesh points in the -0.05-superlevel set of a draw from the Gaussian random field $N(0, (30^2 I - \Delta)^{-1.2})$, where Δ is the Laplacian for functions on the torus.

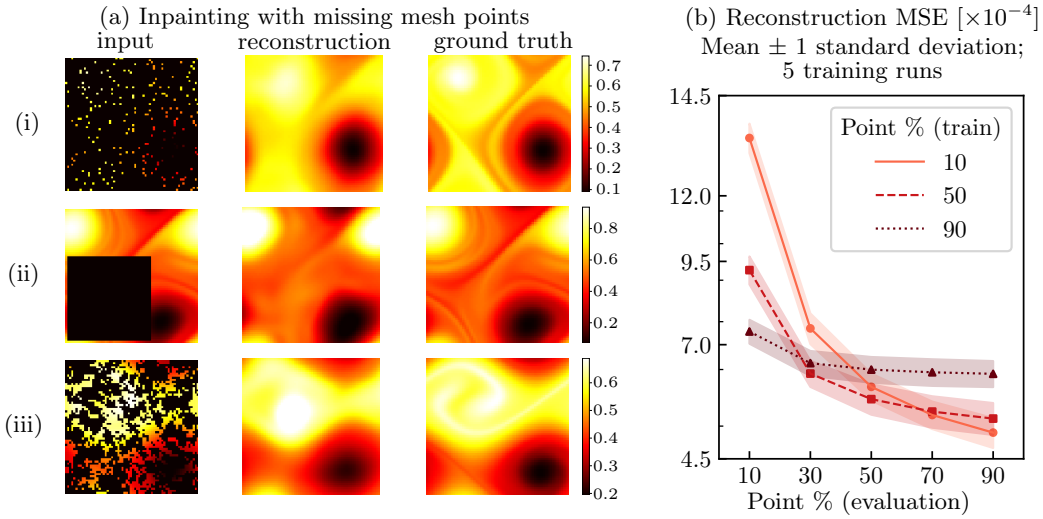


Figure 7: (a) FAE can solve a variety of inpainting tasks; further samples in Appendix B.5. (b) Training the encoder on a sparse mesh has a regularising effect on FAE, leading to lower evaluation MSE on dense meshes, but harms performance on very sparse meshes.

Decoding these samples on a 64×64 grid (Figure 7) leads to reconstructions that agree well with the ground truth, and we find FAE to be robust even with a significant amount of

the original mesh missing. As a consequence of the autoencoding procedure, the observed region of the input may also be modified, an effect most pronounced in (ii), where some features in the input are oversmoothed in the reconstruction. We hypothesise that the failure to capture fine-scale features could be mitigated with better neural-operator architectures.

To understand the effect of the training point ratio on inpainting quality, we first train instances of FAE with complement masking with point ratios 10%, 50%, and 90%. Then, for each model, we evaluate its autoencoding performance by applying random masking to each sample from the held-out set with point ratio $r_{\text{eval}} \in \{10\%, 30\%, 50\%, 70\%, 90\%\}$, reconstructing on the full 64×64 grid, and computing the reconstruction MSE averaged over the held-out set (Figure 7(b)). We observe that the best choice of r_{enc} depends on the point ratio r_{eval} of the input. We hypothesise that when r_{eval} is large, training with r_{enc} small is helpful as training with few mesh points regularises the model. On the other hand, when r_{eval} is small, it is likely that the evaluation mesh is "almost disjoint" from any mesh seen during training, harming performance. Further analysis is provided in Appendix B.5.

Superresolution. The ability to encode and decode on different meshes also enables the use of FAE for single-image superresolution: high-resolution reconstruction of a low-resolution input. Superresolution methods based on deep learning have found use in varied applications including imaging (Li et al., 2024) and fluid dynamics, e.g., in increasing the resolution ("downscaling") of numerical simulations (Kochkov et al., 2021; Bischoff and Deck, 2024). Generalising other continuous superresolution models such as the Local Implicit Image Function (Chen et al., 2021), a single trained FAE model can be applied with any upsampling factor, and FAE has the further advantage of accepting inputs at any resolution.

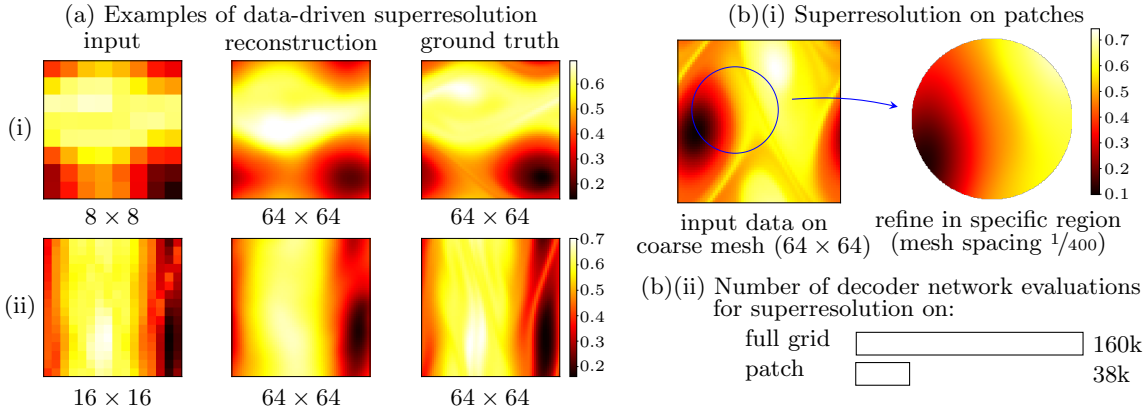


Figure 8: (a) FAE can encode low-resolution inputs and decode at higher resolution, recovering fine-scale features using knowledge of the underlying data. Further examples in Appendix B.5. (b) Evaluating the decoder in a specific subregion can lead to significant computational savings compared to performing superresolution on the full grid.

For **data-driven superresolution**—where we train a model at high resolution and use it to enhance low-resolution inputs at inference time—FAE is able to resolve unseen features from 8×8 and 16×16 inputs on a 64×64 output grid after training at resolution 64×64 (Figure 8(a)). As with inpainting, superresolution performance could be further improved with an architecture that is better able to capture the turbulent dynamics in the data.

We also investigate the stability of FAE for **zero-shot superresolution** (Li et al., 2021), where the model is evaluated on higher resolutions than seen during training. Since FAE is purely data-driven, we view this as a test of the model's mesh-invariance and do not expect to resolve high-frequency features that were not seen during training. Our architecture proves robust when autoencoding on meshes much finer than the original 64×64 training grid (Figure 9(a)); moreover our coordinate MLP architecture allows us to decode on extremely fine meshes without exhausting the GPU memory (details in Appendix B.5). While zero-shot superresolution is possible with VANO when the input is given on the mesh seen during training, FAE can be used for superresolution with any input.

Efficient Superresolution on Regions of Interest. Since our decoder can be evaluated on any mesh, we can perform superresolution in a specific subregion without upsampling across the whole domain. Doing this can significantly reduce inference time, memory usage, and energy cost. As an example, we consider the task of reconstructing a circular subregion of interest with target mesh spacing $1/400$ (Figure 8(b)(i)). Achieving this resolution over the whole domain—corresponding to a 400×400 grid—would involve 160,000 evaluations of the decoder network; decoding on the subregion requires just $1/4$ of this (Figure 8(b)(ii)).

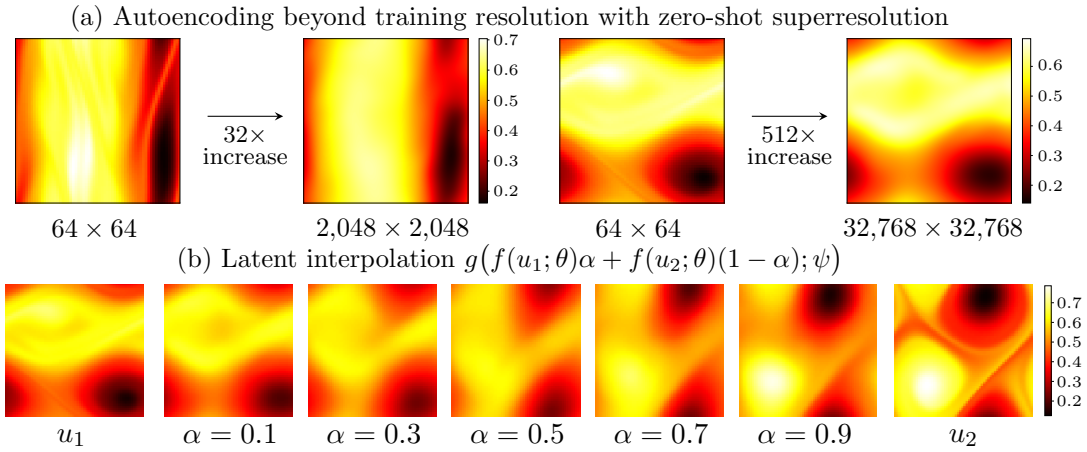


Figure 9: (a) FAE can stably decode at resolutions much higher than the training resolution (best viewed digitally). (b) The regularised latent space $\mathcal{Z}$ allows for meaningful interpolation between samples. Further examples are given in Appendix B.5.

Applications of the Latent Space $\mathcal{Z}$. The regularised FAE latent space gives a well-structured finite representation of the infinite-dimensional data $u \in \mathcal{U}$. We expect there to be benefit in using this representation as a building block for applications such as supervised learning and generative modelling on functional data, similar in spirit to other supervised operator-learning methods with encoder–decoder structure (Seidman et al., 2022).

As a first step towards verifying that the latent space does indeed capture useful structure beyond mere memorisation of the training data, we draw u_1 and u_2 from the held-out set, compute latent vectors $z_1 = f(u_1;\theta)$ and $z_2 = f(u_2;\theta) \in \mathcal{Z}$, and evaluate the decoder g along the convex combination $z_1\alpha + z_2(1-\alpha)$. This leads to a sensible interpolation in $\mathcal{U}$, suggesting that the latent representation is robust and well-regularised (Figure 9(b)).

4.3.2 Darcy Flow

Darcy flow is a model of steady-state flow in a porous medium, derivable from first principles using homogenisation; see, e.g., Freeze and Cherry (1979, Sec. 2.11) and Keller (1980). We restrict attention to the two-dimensional domain $\Omega = [0,1]^2$ and suppose that, for some permeability field $k\colon \Omega \to \mathbb{R}$ and forcing $\varphi\colon \Omega \to \mathbb{R}$, the pressure field $p\colon \Omega \to \mathbb{R}$ satisfies

$$\begin{aligned} -\nabla \cdot \big(k \nabla p\big) &= \varphi \quad \text{on } \Omega, \\ p &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{33}$$

We assume $\varphi = 1$ and that k is distributed as the pushforward of the distribution $N\big(0, (-\Delta + 9I)^{-2}\big)$, where Δ is the Laplacian restricted to functions defined on Ω with zero Neumann data on $\partial\Omega$, under the map $x \mapsto 3 + 9 \cdot \mathbb{1}\big[x \geqslant 0\big]$. We take $\mathcal{U} = L^2(\Omega)$ and define $\Upsilon \in \mathscr{P}(\mathcal{U})$ to be the distribution of pressure fields p solving (33) with permeability k. While solutions to this elliptic PDE can be expected to have greater smoothness (Evans, 2010, Sec. 6.3), we assume only that $p \in L^2(\Omega)$ and use the L^2-norm in the FAE objective (31).

The training data set is based on that of Li et al. (2021) and consists of 1,024 samples from Υ on a 421×421 grid, with a further 1,024 samples held out as an evaluation set. Data are scaled so that $p(x) \in [0,1]$ for all $x \in \Omega$ and, where specified, we downsample as described in Appendix B.6. We train FAE with $d_{\mathcal{Z}} = 64$ and $\beta = 10^{-3}$, and use complement masking with a point ratio r_{enc} of 30%.

Accelerating Training Using Masking. As well as improving reconstructions and robustness to mesh changes, masked training can greatly reduce the cost of training. To illustrate this we compare the training dynamics of FAE on data downsampled to resolution 211×211, using random masking with point ratio $r_{\text{enc}} = r_{\text{dec}} \in \{10\%, 50\%, 90\%\}$. Since the evaluation cost of the encoder and decoder scales linearly with the number of mesh points, we expect significant computational gains when using low point ratios. We perform five training runs for each model and compute the average reconstruction MSE over time on held-out data at resolution 211×211. The models trained with masking converge faster as the smaller data tensors allow for better use of the GPU parallelism (Figure 10). At higher resolutions, memory constraints may preclude training on the full grid, making masking vital.

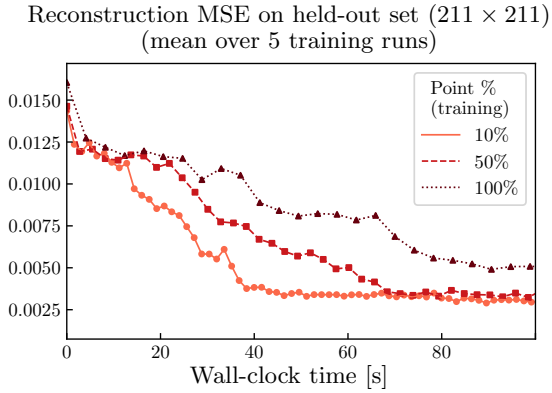


Figure 10: Training with masking in the encoder and decoder reduces training time.

Related ideas are used in the adaptive-subsampling training scheme for FNOs proposed by Lanthaler et al. (2024), which involves training first on a coarse grid and refining the mesh each time the evaluation metric plateaus; our approach differs by dropping mesh points randomly, which would not be possible with FNO. One can readily imagine training FAE with a combination of adaptive subsampling and masking.

Generative Modelling. While FAE is not itself a generative model, it can be made so by training a fixed-dimension generative model on the latent space $\mathcal{Z}$ (Ghosh et al., 2020;

Vahdat et al., 2021). More precisely, we know that applying the FAE encoder f to data induces a distribution $\Sigma^\theta \in \mathscr{P}(\mathcal{Z})$ for z given by

$$z \mid u = f(u;\theta), \quad u \sim \Upsilon. \tag{34}$$

Unlike with FVAE, there is no reason that this should be close to Gaussian. However, we can approximate Σ^θ with a fixed-resolution generative model $\mathbb{P}_z^\varphi \in \mathscr{P}(\mathcal{Z})$ parametrised by $\varphi \in \Phi$, and define the FAE generative model $\mathbb{P}_u^{\psi,\varphi}$ for data u by

$$\text{(FAE generative model)} \qquad u \mid z = g(z;\psi), \quad z \sim \mathbb{P}_z^\varphi. \tag{35}$$

Since applying the decoder to Σ^θ should approximately recover the data distribution if $g(f(u;\theta);\psi) \approx u$ for $u \sim \Upsilon$, we hope that when $\Sigma^\theta \approx \mathbb{P}_z^\varphi$, samples from (35) will be approximately distributed according to Υ. As a simple illustration, we train FAE at resolution 47×47 and fit a Gaussian mixture model $\mathbb{P}_z^\varphi$ with 10 components to Σ^θ using the expectation-maximisation algorithm (see Bishop, 2006, Sec. 9.2.2). Samples from (35) closely resemble those from the held-out data set (Figure 11(a)), and as a result of our mesh-invariant architecture, it is possible to generate new samples on any mesh.

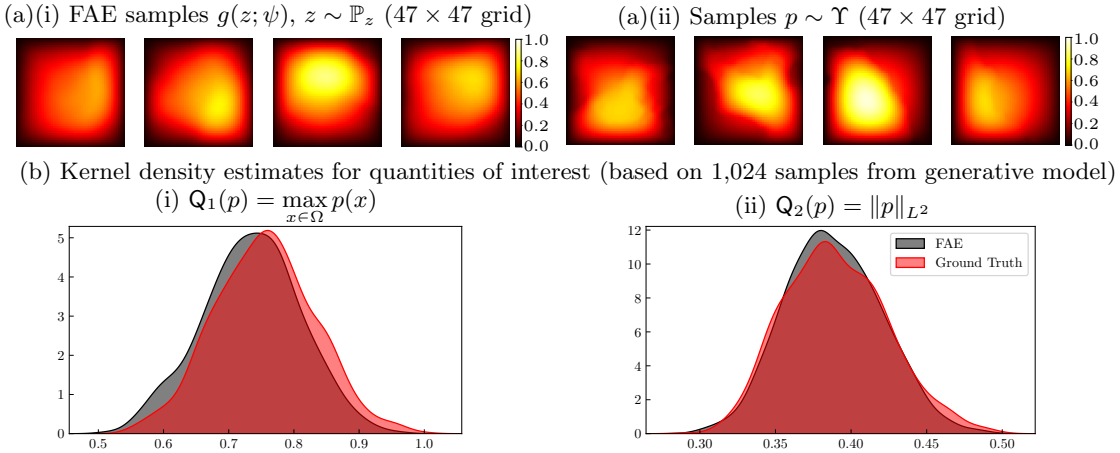


Figure 11: (a) Uncurated samples from the FAE generative model for the pressure field p. Further samples are provided in Appendix B.6. (b) The distributions of quantities of interest computed using the FAE generative model closely agree with the ground truth.

To measure generative performance, we approximate the distributions of physically relevant quantities of interest $\mathsf{Q}_i(p)$ depending on the data $p \in \mathcal{U}$, comparing 1,024 samples from the generative model to the held-out data. Using kernel density estimates as in Figure 4(b), we see close agreement between the distributions (Figure 11(b)). While we could also evaluate the generative model using distances such as maximum mean discrepancy (MMD; Borgwardt et al., 2006), we focus on interpretable quantities relevant to the physical system at hand.

Though we adopt the convention of training the autoencoder and generative model separately (Rombach et al., 2022) here, the models could also be trained jointly; we leave this, and an investigation of generative models on the FAE latent space, to future work.

5. Related Work

Variational Autoencoding Neural Operators. The VANO model (Seidman et al., 2023) was the first to attempt systematic extension of the VAE objective to function space. The paper uses ideas from operator learning to construct a model that can decode—but not encode—at any resolution. Our approach differs in both training objective and practical implementation, as we now outline.

The most significant difference between what is proposed in this paper and in VANO is the objective on function space: the VANO objective coincides with a specific case of our model (16a)–(16d) with the decoder noise $\mathbb{P}_\eta$ being white noise on $L^2([0,1]^d)$. As a consequence the generative model for VANO takes values in $\mathcal{U} = H^s([0,1]^d)$ if and only if $s < -d/2$; in particular generated draws are not in $L^2([0,1]^d)$. Unlike our approach, VANO aims to maximise an extension of the ELBO (15b), in which a regularisation parameter $\beta > 0$ is chosen as a hyperparameter, and the ELBO takes the form

$$\begin{aligned}\mathrm{ELBO}_\beta^{\mathrm{VANO}}(u;\theta,\psi) &= \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\left[\log \frac{\mathrm{d}\mathbb{P}^\psi_{u|z}}{\mathrm{d}\mathbb{P}_\eta}(u)\right] - \beta D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}_z\big) \\ &= \mathop{\mathbb{E}}_{z\sim\mathbb{Q}^\theta_{z|u}}\left[\langle g(z;\psi), u\rangle_{L^2}^{\sim} - \tfrac{1}{2}\|g(z;\psi)\|_{L^2}^2\right] - \beta D_{\mathrm{KL}}\big(\mathbb{Q}^\theta_{z|u} \,\big\|\, \mathbb{P}_z\big),\end{aligned}$$

where the second equality comes from the Cameron–Martin theorem as in Example 2. Maximising $\mathrm{ELBO}_\beta^{\mathrm{VANO}}$ with $\beta = 1$ is precisely equivalent to minimising the per-sample loss (29) from Example 2, so naive application of $\mathrm{ELBO}_\beta^{\mathrm{VANO}}$ will result in the same issues seen there. In particular, discretisations of the ELBO may diverge as resolution is refined; moreover, for data with L^2-regularity, the generative model $\mathbb{P}_u^\psi$ is greatly misspecified, with draws $g(z;\psi) + \eta$, $z \sim \mathbb{P}_z$, $\eta \sim \mathbb{P}_\eta$, lying in a Sobolev space of lower regularity than the data. This issue is obscured by the convention in the VAE literature of considering only the decoder mean $g(z;\psi)$; considering the full generative model with draws $g(z;\psi)+\eta$ reveals the incompatibility more clearly. We argue that the empirical success of VANO in autoencoding is because the objective can be seen as that of a regularised autoencoder (Remark 22).

Along with the differences in the training objective and its interpretation, FVAE differs greatly from VANO in architecture. While the VANO decoders can be discretised on any mesh—and our decoder closely resembles VANO's nonlinear decoder—its encoders assume a fixed mesh for training and inference. In contrast, our encoder can be discretised on any mesh, enabling many of our contributions, such as masked training, inpainting, and superresolution, which are not possible within VANO.

Generative Models on Function Space. Aside from VANO, recent years have seen significant interest in the development of generative models on function space. Several extensions of score-based (Song et al., 2021) and denoising diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) to function space have been proposed (e.g., Pidstrigach et al., 2023; Hagemann et al., 2023; Lim et al., 2023; Kerrigan et al., 2023; Franzese et al., 2023; Zhang and Wonka, 2024). Rahman et al. (2022) propose the generative adversarial neural operator (GANO), extending Wasserstein generative adversarial neural networks (Arjovsky et al., 2017) to function space with FNOs in the generator and discriminator to achieve resolution-invariance.

Variational Inference on Function Space. In machine learning, variational inference on function space also arises in the context of Bayesian neural networks (Sun et al., 2019; Burt et al., 2021; Cinquin and Bamler, 2024). In this setting one wishes to minimise the KL divergence between the posterior in function space and a computationally tractable approximation—but, as in our study, this divergence may be infinite owing to a lack of absolute continuity between the two distributions.

Learning on Point Clouds. Our architecture takes inspiration from the literature on machine learning on point clouds, where data are viewed as sets of points with arbitrary cardinality. Several models for autoencoding and generative modelling with point clouds have been proposed, such as energy-based processes (Yang et al., 2020) and SetVAE (Kim et al., 2021); our work differs by defining a loss in function space, ensuring that our model converges to a continuum limit as the mesh is refined. Continuum limits of semisupervised algorithms for graphs and point clouds have also been studied (e.g., Dunlop et al., 2020).

6. Outlook

Our study of autoencoders on function space has led to FVAE, an extension of VAEs which imposes stringent requirements on the data distribution in infinite dimensions but benefits from firm probabilistic foundations; it has also led to the non-probabilistic FAE, a regularised autoencoder which can be applied much more broadly to functional data.

Benefits. Both FVAE and FAE offer significant benefits when working with functional data, such as enabling training with data across resolutions, inpainting, superresolution, and generative modelling. These benefits are possible only through our pairing of a well-defined objective in function space with mesh-invariant encoder and decoder architectures.

Limitations. FVAE can be applied only when the generative model is sufficiently compatible with the data distribution—a condition that is difficult to satisfy in infinite dimensions, and restricts the applicability to FVAE to specific problem classes. FAE overcomes this restriction, but does not share the probabilistic foundations of FVAE.

The desire to discretise the encoder and decoder on arbitrary meshes rules out many high-performing grid-based architectures, including convolutional networks and FNOs. We believe this is a limiting factor in the numerical experiments, and that combining our work with more complex operator architectures (e.g., Kovachki et al., 2023) or continuum extensions of point-cloud CNNs (Li et al., 2018) would yield further improvements.

Future Work. Our work gives new methods for nonlinear dimension reduction in function space, and we expect there to be benefit in building operator-learning methods that make use of the resulting latent space, in the spirit of PCA-Net (Bhattacharya et al., 2021). For FAE, which unlike FVAE is not inherently a generative model, we expect particular benefit in the use of more sophisticated generative models on the latent space, for example diffusion models, analogous to Stable Diffusion (Rombach et al., 2022).

While our focus has been on scientific problems with synthetic data, our methods could also be applied to real-world data, for example in computer vision; for these challenging data sets, further research on improved mesh-invariant architectures will be vital. Our study has also focussed on the typical machine-learning setting of a fixed dataset of size N; research

into the behaviour of FVAE and FAE in the infinite-data limits using tools from statistical learning theory would also be of interest.

Acknowledgments

JB is supported by Splunk Inc. MG is supported by a Royal Academy of Engineering Research Chair, and Engineering and Physical Sciences Research Council (EPSRC) grants EP/T000414/1, EP/W005816/1, EP/V056441/1, EP/V056522/1, EP/R018413/2, EP/R034710/1, and EP/R004889/1. HL is supported by the Warwick Mathematics Institute Centre for Doctoral Training and gratefully acknowledges funding from the University of Warwick and the EPSRC (grant EP/W524645/1). AMS is supported by a Department of Defense Vannevar Bush Faculty Fellowship and by the SciAI Center, funded by the Office of Naval Research (ONR), under grant N00014-23-1-2729.

Appendix A. Supporting Results

In the following proof we use the fact that the norm of the Sobolev space $H^s([0,1])$ can be written as a weighted sum of frequencies (see Krein and Petunin, 1966, Sec. 9):

$$\|u\|^2_{H^s([0,1])} = \sum_{j\in\mathbb{N}} \big(1+j^2\big)^s |\alpha_j|^2, \quad u = \sum_{j\in\mathbb{N}} \alpha_j e_j, \quad e_j(x) = \sqrt{2}\sin(\pi j x). \tag{36}$$

Proof of Proposition 15 Let η be an L^2-white noise, let $h = \sum_{j\in\mathbb{N}} h_j e_j \in L^2([0,1])$, and note that $\mathbb{P}_\eta(\cdot - h)$ is the distribution of the random variable $\eta + h$. Thus, writing out the H^s-norm using the Karhunen–Loève expansion of η, we see that

$$\|\eta + h\|^2_{H^s([0,1])} = \sum_{j\in\mathbb{N}} \big(1+j^2\big)^s \big|\xi_j + h_j\big|^2. \tag{37}$$

First, we show that $\|\eta + h\|_{H^s([0,1])} < \infty$ almost surely when $s < -1/2$. To do this we apply the Kolmogorov two-series theorem (Durrett, 2019, Theorem 2.5.6), which states that the random series (37) converges almost surely if

$$\sum_{j\in\mathbb{N}} \big(1+j^2\big)^s \mathbb{E}\Big[\big(\xi_j + h_j\big)^2\Big] < \infty \quad \text{and} \quad \sum_{j\in\mathbb{N}} \big(1+j^2\big)^{2s} \operatorname{Var}\Big(\big(\xi_j + h_j\big)^2\Big) < \infty.$$

But, since $\xi_j \sim N(0,1)$, we know that $\mathbb{E}[\xi_j^2] = 1$ and $\operatorname{Var}(\xi_j^2) = 2$; applying this, the elementary identity $(x+y)^2 \leqslant 2x^2 + 2y^2$ for $x, y \in \mathbb{R}$, and the fact that $\sum_{j\in\mathbb{N}} j^\alpha < \infty$ for $\alpha < -1$ shows that the two series are finite. To see that $\mathbb{P}_\eta(\cdot - h)$ assigns zero probability to $L^2([0,1])$, suppose for contradiction that $\eta + h$ had finite L^2-norm; then η would also have finite L^2-norm. But as a consequence of the Borel–Cantelli lemma,

$$\big\|\eta\big\|^2_{L^2([0,1])} = \sum_{j\in\mathbb{N}} \xi_j^2 = \infty \ \text{ almost surely,}$$

because the summands are independent and identically distributed, and thus for any constant $c > 0$, infinitely many summands exceed c with probability one. ■

Lemma 25 *Suppose that $\mathcal{U} = C_0([0,T],\mathbb{R}^m)$ and that $\mu \in \mathscr{P}(\mathcal{U})$ and $\nu \in \mathscr{P}(\mathcal{U})$ are the laws of the $\mathbb{R}^m$-valued diffusions*

$$\begin{aligned} \mathrm{d}u_t &= b(u_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \qquad u_0 = 0, \qquad t \in [0,T] \\ \mathrm{d}v_t &= c(v_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t, \qquad v_0 = 0, \qquad t \in [0,T]. \end{aligned}$$

where $(w_t)_{t\in[0,T]}$ is a Brownian motion on $\mathbb{R}^m$. Suppose that the Novikov condition (19) *holds for both processes. Then*

$$D_{\mathrm{KL}}(\mu \,\|\, \nu) = \mathop{\mathbb{E}}_{u\sim\mu}\left[\frac{1}{2\varepsilon}\int_0^T \|b(u_t) - c(u_t)\|_2^2\,\mathrm{d}t\right].$$

Proof Applying the Girsanov formula (20) to obtain the density $\mathrm{d}\mu/\mathrm{d}\nu$, taking logarithms to evaluate $D_{\mathrm{KL}}(\mu\,\|\,\nu)$, and noting that under μ we have $\mathrm{d}u_t = b(u_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}w_t$, we obtain

$$D_{\mathrm{KL}}(\mu \,\|\, \nu) = \mathop{\mathbb{E}}_{u\sim\mu}\left[\frac{1}{2\varepsilon}\int_0^T \|b(u_t) - c(u_t)\|_2^2\,\mathrm{d}t - \frac{1}{\sqrt{\varepsilon}}\int_0^T \langle b(u_t) - c(u_t),\, \mathrm{d}w_t\rangle\right].$$

Under μ, the process $(w_t)_{t\in[0,T]}$ is Brownian motion and so the second expectation is zero. ■

Appendix B. Experimental Details

In this section, we provide additional details, training configurations, samples, and analysis for the numerical experiments in Section 2.5 and Section 4.3. All experiments were run on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of VRAM.

B.1 Base Architecture

We use the common architecture described in Section 2.4 and Section 4.2 for all experiments, using the Adam optimiser (Kingma and Ba, 2015) with the default hyperparameters ε, β_1, and β_2; we specify the learning rate and learning-rate decay schedule for each experiment in what follows.

Positional Encodings. Where specified, both the encoder and decoder will make use of Gaussian random Fourier features (Tancik et al., 2020), pairing the query coordinate $x \in \Omega \subset \mathbb{R}^d$ with a positional encoding $\gamma(x) \in \mathbb{R}^{2k}$. To generate these encodings, a matrix $B \in \mathbb{R}^{k \times d}$ with independent $N(0, I)$ entries is sampled and viewed as a hyperparameter of the model to be used in both the encoder and decoder. The positional encoding $\gamma(x)$ is then given by the concatenated vector $\gamma(x) = \left[\cos(2\pi Bx); \sin(2\pi Bx)\right]^T \in \mathbb{R}^{2k}$ where the sine and cosine functions are applied componentwise to the vector $2\pi Bx$.

B.2 Brownian Dynamics

The training data consists of 8,192 samples from the path distribution Υ of the SDE (27),(28) on the time interval $[0, T]$, $T = 5$. Trajectories are generated using the Euler–Maruyama scheme with internal time step $1/8{,}192$ (unrelated to the choice to take 8,192 training samples), and the resulting paths are then subsampled by a factor of 80 to obtain the training data. Thus the data have effective time increment $5/512$; moreover the path information is removed at 50% of the points resulting from these time increments, chosen uniformly at random.

Experimental Setup. We train for 100,000 steps with initial learning rate 10^{-3} and an exponential decay of 0.98 applied every 1,000 steps, with batch size 32 and 4 Monte Carlo samples for $\mathbb{Q}^{\theta}_{z|u}$. We use latent dimension $d_{\mathcal{Z}} = 1$, $\beta = 1.2$ and $\lambda = 10$. The three sets of simulations shown in Figure 2 use $\kappa = 0$, 25, and 10,000 respectively.

B.3 Estimation of Markov State Models

Data and Discretisation. To validate the ability of FVAE to model higher-dimensional SDE trajectories, we specify a simple potential with qualitative features similar to those arising in the complex potential surfaces arising in molecular dynamics. To this end, define the centres $c_1 = (0, 0)$, $c_2 = (0.2, 0.2)$, $c_3 = (-0.2, -0.2)$, $c_4 = (0.2, -0.2)$, $c_5 = (0, 0.2)$ and $c_6 = (-0.2, 0)$; standard deviations $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 0.1$ and $\sigma_5 = \sigma_6 = 0.03$; and masses $m_1 = m_2 = m_3 = m_4 = 0.1$ and $m_5 = m_6 = 0.01$. Then let

$$U(x) = 0.3\left[0.5(x_1 + x_2) + x_1^2 + x_2^2 - \sum_{i=1}^{6} m_i N\left(x; c_i, \sigma_i^2 I_2\right)\right].$$

This potential has three key components: a linear term breaking the symmetry, a quadratic term preventing paths from veering too far from the path's starting point, the origin,

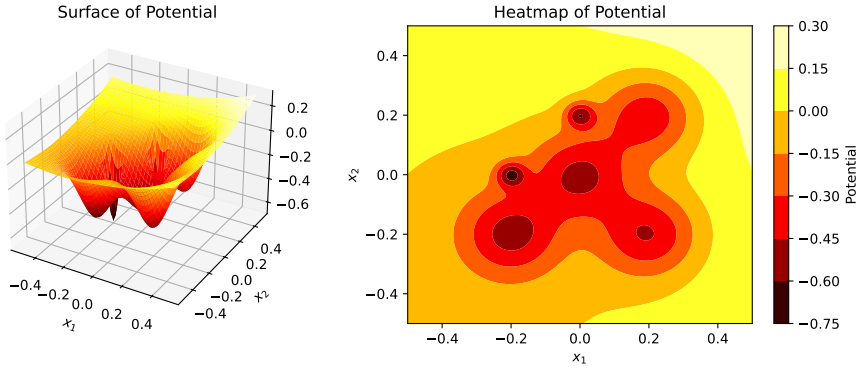


Figure 12: Potential function $U\colon \mathbb{R}^2 \to \mathbb{R}$ for Section 2.5.2.

and negative Gaussian densities—serving as potential wells—positioned at the centres c_i (Figure 12). Sample paths of (27) with initial condition $u_0 = 0$, temperature $\varepsilon = 0.1$ and final time $T = 3$ show significant diversity, with many paths transitioning at least once between different wells (see ground truth in Figure 13).

Experimental Setup. The training set consists of 16,384 paths generated with an Euler–Maruyama scheme with internal time step $1/8{,}192$, subsampled by a factor 48 to obtain an equally spaced mesh of 513 points. We take $d_{\mathcal{Z}} = 16$, $\beta = 10$, $\kappa = 50$, and $\lambda = 50$, and, as in Appendix B.3, train on data where 50% of the points on the path are missing. We also use the same learning rate, learning-rate decay schedule, step limit, and batch size.

Results. FVAE's reconstructions closely match the inputs (Figure 13), and FVAE produces convincing generative samples capturing qualitative features of the data (Figure 14).

B.4 Dirac Distributions

Data and Discretisation. We view Υ as a probability distribution on $\mathcal{U} = H^{-1}([0,1])$. At each resolution I, we discretise the domain $[0,1]$ using an evenly spaced mesh of points $\{i/I+1\}_{i=1,\dots,I}$ and approximate the Dirac mass δ_ξ, $\xi \in [0,1]$, by the optimal L^1-approximation: a discretised function which is zero except at the mesh point closest to ξ, normalised to have unit L^1-norm. The training data set consists of discretised Dirac functions at each mesh point; the goal is not to train a practical model for generalisation, but to isolate the effect of the objective.

Experimental Setup. We train FVAE and FAE models at resolutions $I \in \{8, 16, 32, 64, 128\}$. For each model, we perform 50 independent runs of 30,000 steps with batch size 6.

Architecture. The neural network $\rho\colon \mathbb{R} \times \Theta \to \mathbb{R} \times \mathbb{R}$ in the encoder map f is assumed to have 3 hidden layers of width 128, and the mean $\mu(z;\psi)$ and standard deviation $\sigma(z;\psi)$ in the decoder are computed from a 3-layer neural network of width 128. For numerical stability, we impose a lower bound on σ based on the mesh spacing Δx, given by $\sigma_{\min}(\Delta x) = (2\pi)^{-1/2}\Delta x$.

FVAE Configuration. We view data $u \sim \Upsilon$ as lying in the Sobolev space $\mathcal{U} = H^{-1}([0,1])$; the decoder g will output functions in $L^2([0,1])$ and we take decoder-noise distribution

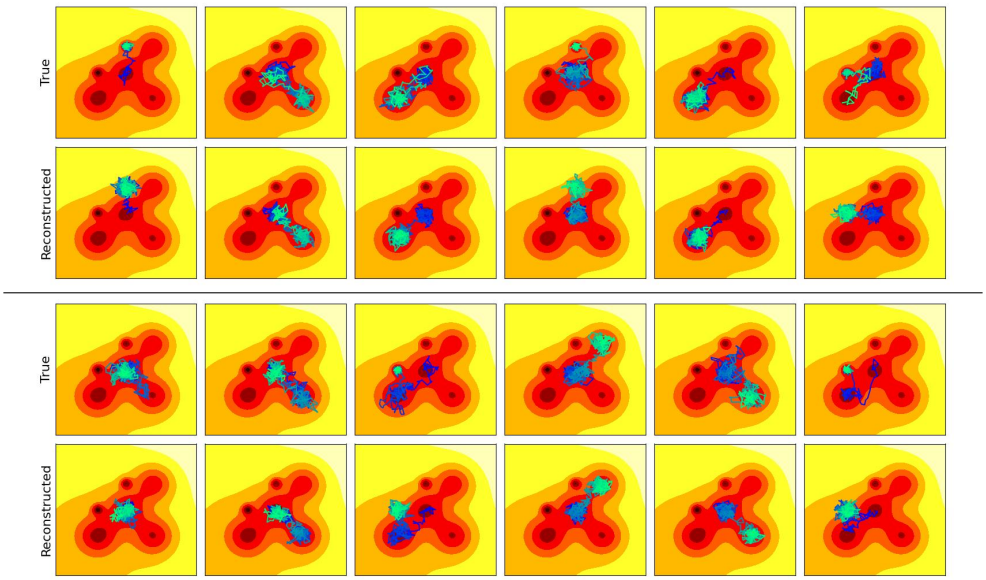
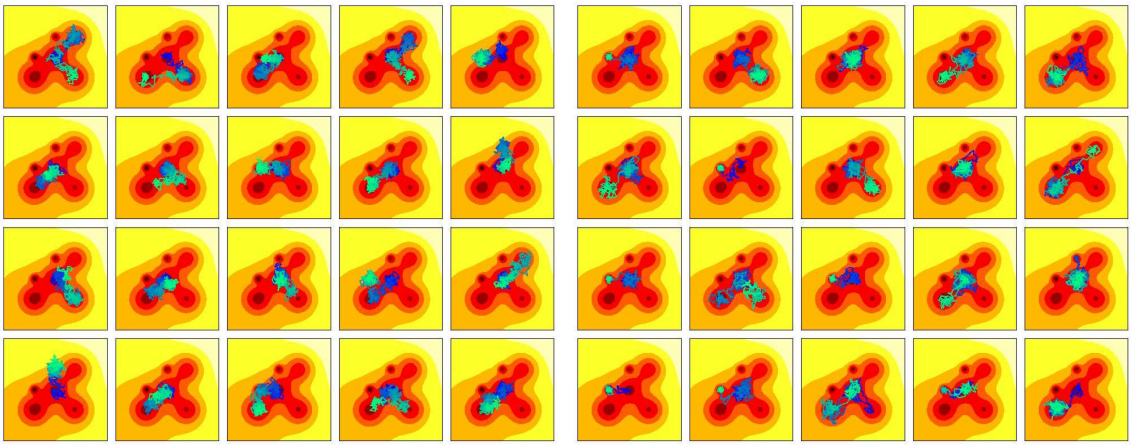


Figure 13: Held-out ground-truth data from the SDE in Section 2.5.2 ("True" row) and the corresponding FVAE reconstructions of sample paths ("Reconstructed" row).



(a) FVAE (b) Data set

Figure 14: (a) Samples of the SDE in Section 2.5.2 drawn from the FVAE generative model with randomly drawn latent vector $z \sim \mathbb{P}_z$. (b) Ground-truth paths of the SDE in Section 2.5.2 generated using an Euler–Maruyama solver. In both subfigures, the evolution through time $t \in [0, 3]$ is depicted as a transition in colour from blue to green.

$\mathbb{P}_\eta = N(0, I)$, noting that $\mathbb{P}_\eta \in \mathscr{P}(H^s([0,1]))$ if and only if $s < -1/2$; in particular white-noise samples do not lie in the space $L^2([0,1])$. We modify the per-sample loss (7b) by reweighting the term $D_{\mathrm{KL}}(\mathbb{Q}^\theta_{z|u} \,\|\, \mathbb{P}_z)$ by $\beta = 10^{-4}$, and take 16 Monte Carlo samples for $\mathbb{Q}^\theta_{z|u}$. We use an initial learning rate of 10^{-4}, decaying exponentially by a factor 0.7 every 1,000 steps.

FAE Configuration. To compute the H^{-1}-norm we truncate the series expansion (36) and compute coefficients α_j from a discretisation of u using the discrete sine transform. We use initial learning rate 10^{-4}, decaying exponentially by a factor 0.9 every 1,000 steps, and take $\beta = 10^{-12}$. For consistency with the FVAE loss, we subtract the squared data norm $\frac{1}{2}\|u\|^2_{H^{-1}}$ from the FAE loss, yielding the expression

$$\frac{1}{2}\big\|g(f(u;\theta);\psi) - u\big\|^2_{H^{-1}} - \frac{1}{2}\|u\|^2_{H^{-1}} = \frac{1}{2}\left\|g\big(f(u;\theta);\psi\big)\right\|^2_{H^{-1}} - \left\langle g\big(f(u;\theta);\psi\big), u\right\rangle_{H^{-1}}.$$

Results. As expected, the final training loss under both models decreases as the resolution is refined, since the lower bound $\sigma_{\min}$ decreases. However, the FAE loss appears to converge and is stable across runs, while the FVAE loss appears to diverge and becomes increasingly unstable across runs. This gives convincing empirical evidence that the joint divergence (5) for FVAE is not defined as a result of the misspecified decoder noise; the use of FAE with an appropriate data norm alleviates this issue. Since the FVAE objective with $\mathbb{P}_\eta = N(0, I)$ coincides with the VANO objective, this issue would also be present for VANO. Under both models, training becomes increasingly unstable at high resolutions: when σ is small, the loss becomes highly sensitive to changes in μ; this instability is unrelated to the divergence of the FVAE training loss and is a consequence of training through gradient descent.

B.5 Incompressible Navier–Stokes Equations

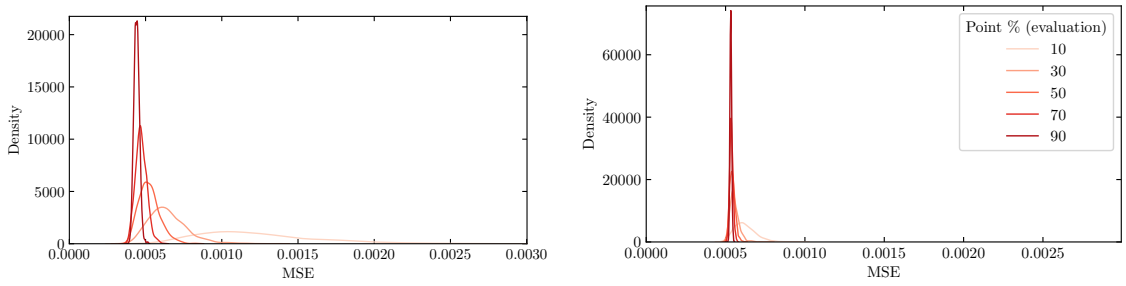
Viscosity ν	Resolution	Train Samples	Eval. Samples	Snapshot Time T
10^{-3}	64×64	4,000	1,000	50
10^{-4}	64×64	8,000	2,000	50
10^{-5}	64×64	960	240	20

Table 2: Details of Navier–Stokes data sets.

Data and Discretisation. We use data as provided online by Li et al. (2021). Solutions of (32) are generated by sampling the initial condition from the Gaussian random field $N(0, C)$, $C = 7^{3/2}(49I - \Delta)^{-5/2}$, and evolving in time using a pseudospectral method. While the data of Li et al. (2021) includes the full time evolution, we use only snapshots of the vorticity at the final time. Every snapshot is a 64×64 image, normalised to take values in $[0, 1]$; details of this data set are given in Table 2.

Effects of Point Ratios. Here, we extend the analysis of Figure 7(b) to understand how the point ratio used during training affects reconstruction performance. We first train two FAE models on the Navier–Stokes data set with viscosity $\nu = 10^{-4}$, using complement masking with a point ratio r_{enc} of 10% and 90% respectively. Then, we fix an arbitrary sample from the held-out set and, for each model, generate 1,000 distinct masks with point ratios 10%, 30%, 50%, 70%, and 90%. We then encode on each mesh and decode on the full grid and compute kernel density estimates of the reconstruction MSE (Figure 15). The model trained with $r_{\text{enc}} = 10\%$ is much more sensitive to the location of the evaluation mesh points, especially when the evaluation point ratio is low; with sufficiently high encoder point ratio at evaluation time, however, the reconstruction MSE of the model trained using $r_{\text{enc}} = 10\%$

surpasses that of the model trained at $r_{\text{enc}} = 90\%$. This suggests a tradeoff whereby a higher training point ratio provides more stability, at the cost of increasing autoencoding MSE, particularly when the point ratio of the evaluation data is high. We hypothesise that a lower training ratio regularises the model to attain a more robust internal representation.



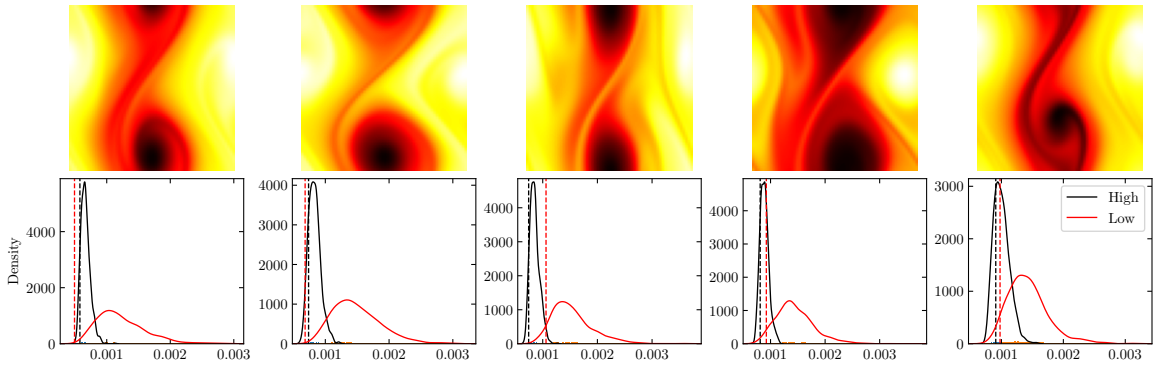
(a) For a model trained with point ratio 10%. (b) For a model trained with a point ratio 90%.

Figure 15: Kernel density estimates for full-grid reconstruction MSE on the reference sample across 1,000 randomly chosen meshes. Training with a low point ratio regularises, reducing MSE when the evaluation data has a high point ratio, but at the cost of greater variance when evaluating on low point ratios.

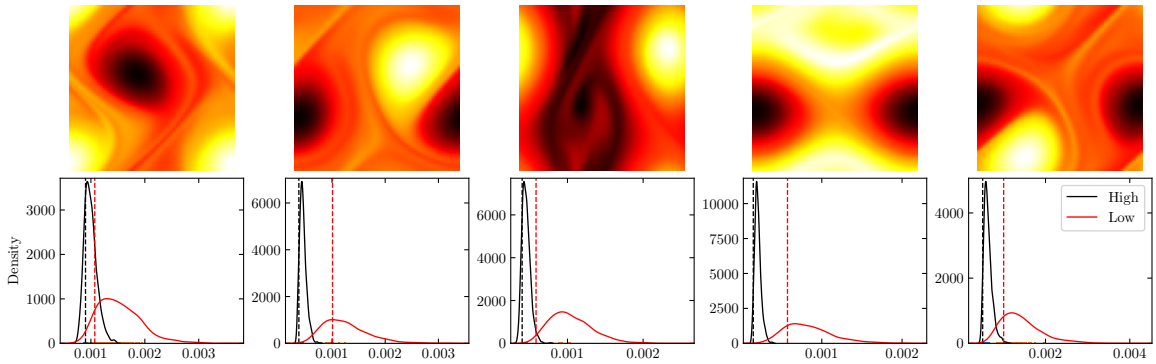
We also investigate the sensitivity of the models to a specific encoder mesh, seeking to understand whether an encoder mesh achieving low MSE on one image leads to low MSE on other images. The procedure is as follows: we select an image arbitrarily from the held-out set (the **reference sample**) and draw 1,000 random meshes with point ratio 10%; then, we select the mesh resulting in the lowest reconstruction MSE for each of the two models. For the nearest neighbours of the chosen sample in the held-out set, the reconstruction error on this MSE-minimising mesh is lower than average (Figure 16(a); dashed lines), suggesting that a good configuration will yield good results on similar samples. On the other hand, using the MSE-minimising mesh on arbitrary samples from the held-out set yields an MSE somewhat lower than a randomly chosen mesh; unsurprisingly, however, the arbitrarily chosen samples appear to benefit less than the nearest neighbours (Figure 16(b)).

Experimental Setup. We train for 50,000 steps with batch size 32 and initial learning rate 10^{-3}, decaying exponentially by a factor 0.98 every 1,000 steps. We use complement masking with $r_{\text{enc}} = 0.3$, providing a good balance of performance and robustness to masking.

Architecture. Both the CNN and FAE architecture use Gaussian random positional encodings with $k = 16$. For the sake of comparison, we use a standard CNN architecture inspired by the VGG model (Simonyan and Zisserman, 2015), gradually contracting/expanding the feature map while increasing/decreasing the channel dimensions. The architecture we use was identified using a search over parameters such as the network depth while maintaining a similar parameter count to our baseline FAE model. The encoder consists of four CNN layers with output channel dimension 4, 4, 8, and 16 respectively and kernel sizes are 2, 2, 4, and 4 respectively, all with stride 2. The result is flattened and passed through a single-hidden-layer MLP of width 64 to obtain a vector of dimension 64. The decoder consists of a single-layer MLP of width 64 and output dimension 512, which is then rearranged to a



(a) Nearest neighbours of the reference sample in the held-out set.



(b) Arbitrarily chosen samples from the held-out set.

Figure 16: Kernel density estimates of full-grid reconstruction MSE for models trained at 10% (Low) and 90% (High) point ratios on samples from the held-out set. Dashed lines indicate the MSE obtained using the mesh minimising MSE on the reference sample.

4×4 feature map with channel size 32. This feature map is then passed through four layers of transposed convolutions that respectively map to 16, 8, 4, and 4 channel dimensions, with kernel sizes 4, 4, 2, and 2 respectively, and stride 2. The result is then mapped by two CNN layers with kernel size 3, stride 1, and output channel dimension 8 and 1 respectively.

Uncurated Reconstructions and Samples. Reconstructions of randomly selected data from the held-out sets for viscosities $\nu = 10^{-3}$, 10^{-4} and 10^{-5} are provided in Figures 17, 18, and 19 respectively. As described in Section 4.3.2, we apply FAE as a generative model by fitting a Gaussian mixture with 10 components on the latent space. Samples from models trained at $\nu = 10^{-3}$, 10^{-4} and 10^{-5} are shown in Figures 20, 21, and 22 respectively.

Evaluation at Very High Resolutions. In Figure 9, we demonstrate zero-shot resolution by evaluating the decoder on grids of resolution $2{,}048 \times 2{,}048$ and $32{,}768 \times 32{,}768$. While the former requires approximately 16 MB to store using 32-bit floating-point numbers, the latter requires 4.3 GB, and thus applying a neural network directly to the $32{,}768 \times 32{,}768$ image is more likely to exhaust GPU memory. To allow evaluation of the decoder at this resolution, we partition the domain into 1,000 chunks and evaluate the decoder on each chunk in turn; we then reassemble the resulting data in the RAM. To ensure that each

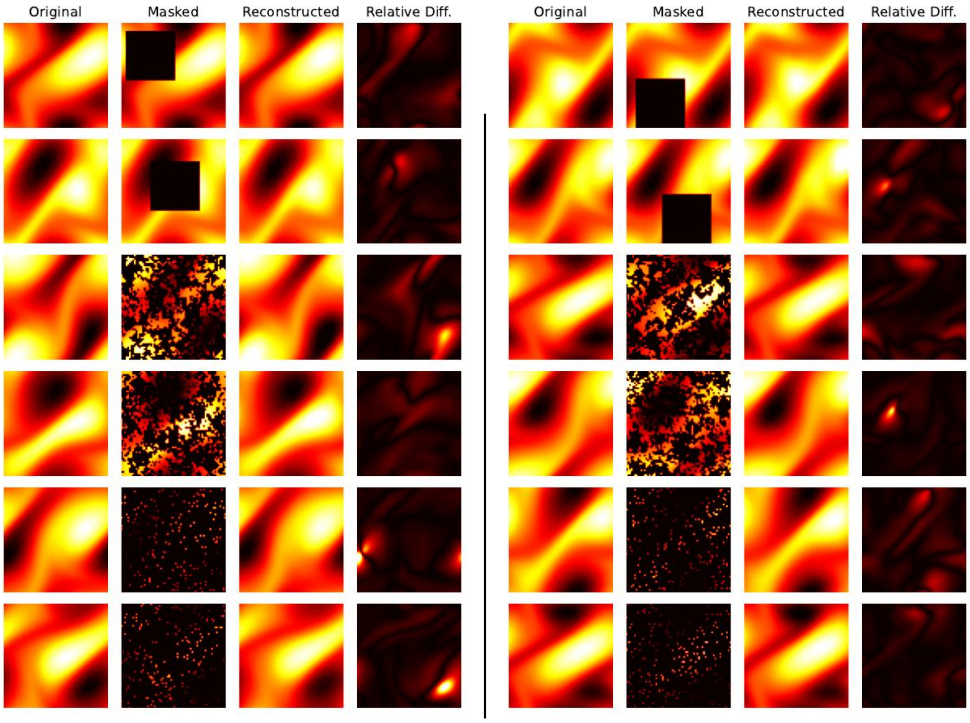


Figure 17: FAE reconstructions of Navier–Stokes data with viscosity 10^{-3}.

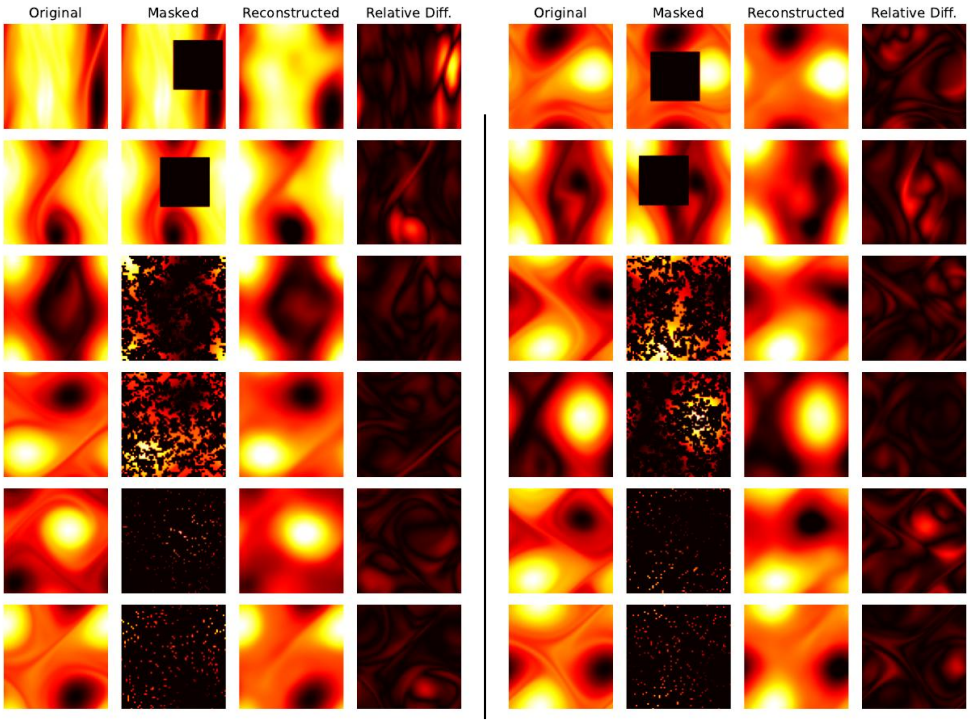


Figure 18: FAE reconstructions of Navier–Stokes data with viscosity 10^{-4}.

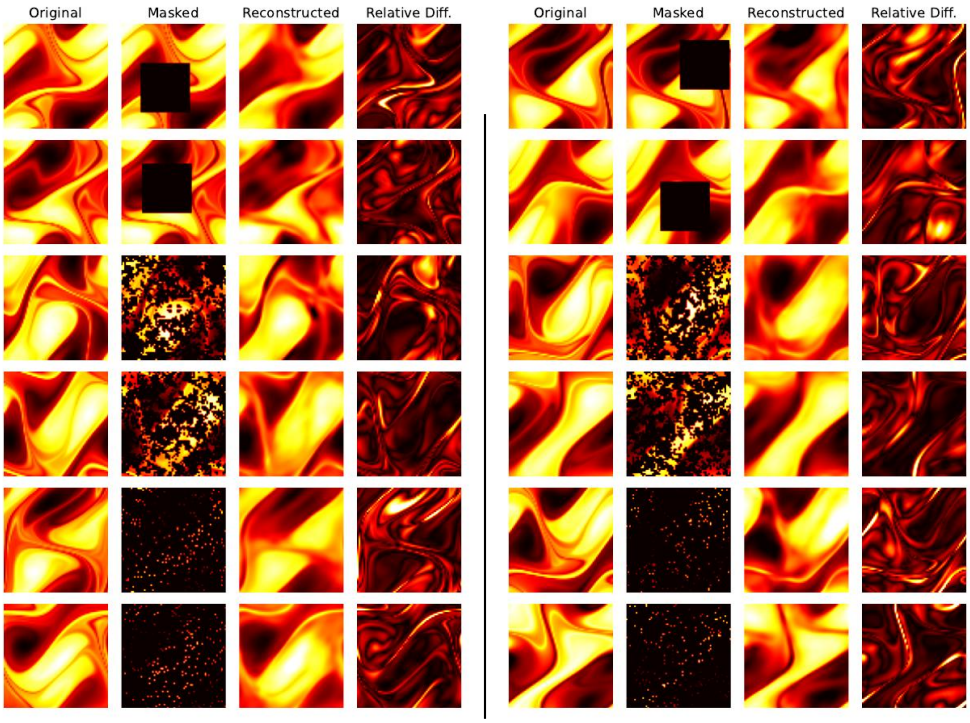
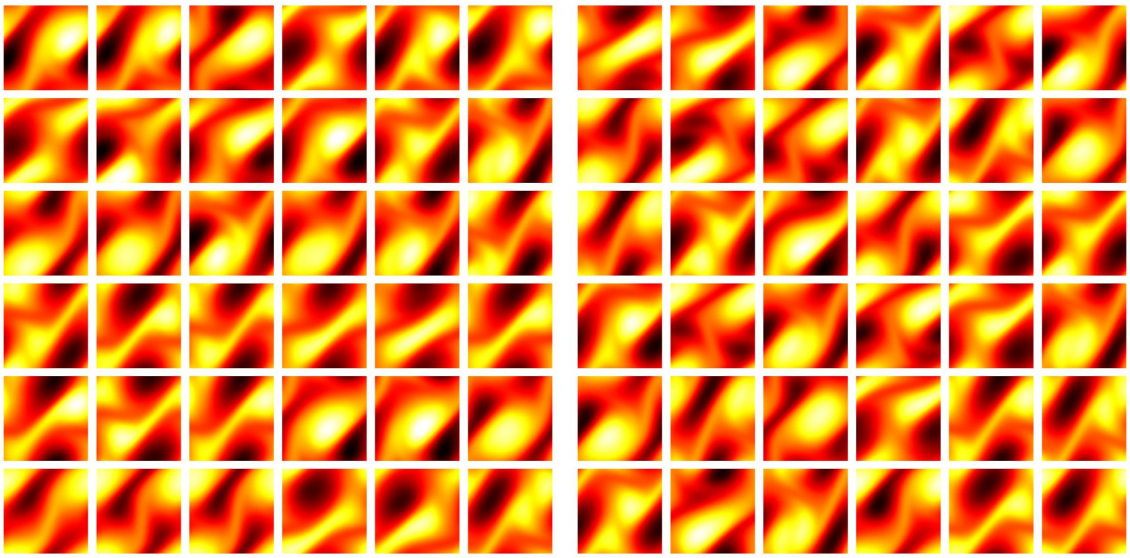
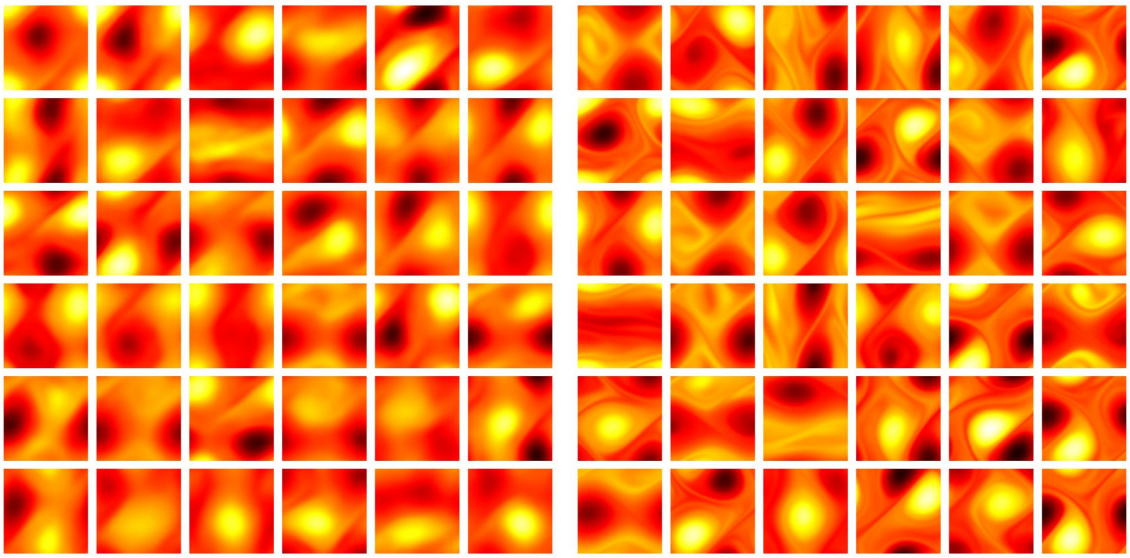


Figure 19: FAE reconstructions of Navier–Stokes data with viscosity $\nu = 10^{-5}$.



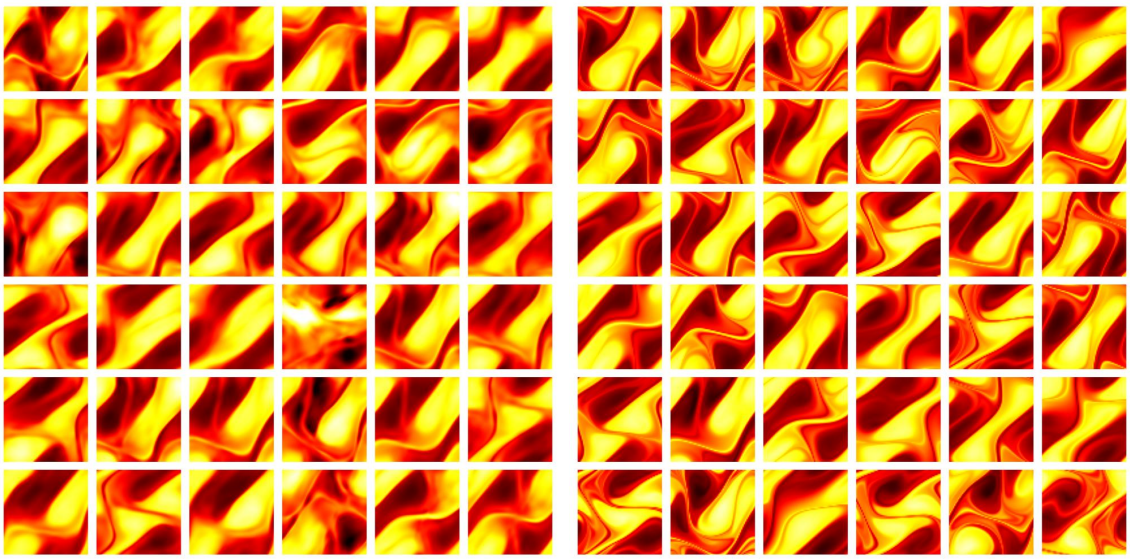
(a) FAE (b) Data set

Figure 20: Samples of Navier–Stokes data with viscosity $\nu = 10^{-3}$.



(a) FAE (b) Data set

Figure 21: Samples of Navier–Stokes data with viscosity $\nu = 10^{-4}$.



(a) FAE (b) Data set

Figure 22: Samples of Navier–Stokes data with viscosity $\nu = 10^{-5}$.

chunk has an integer number of points, we take the first 824 chunks to contain 1,073,742 mesh points ($\approx$ 4 MB), and take the remaining 176 chunks to contain 1,073,741 points.

B.6 Darcy Flow

Data Set. The data we use is based on that provided online by Li et al. (2021), given on a 421×421 grid and generated through a finite-difference scheme. Where described, we downsample this data to lower resolutions by applying a low-pass filter in Fourier space and subsampling the resulting image. The low-pass filter is a mollification of an ideal sinc filter with bandwidth selected to eliminate frequencies beyond the Nyquist frequency of the target resolution, computed by convolving the ideal filter in Fourier space with a Gaussian kernel with standard deviation $\sigma = 0.1$, truncated to a 7×7 convolutional filter.

Experimental Setup. We follow the same setup used for the Navier–Stokes data set: we train for 50,000 steps, with batch size 32 and complement masking with $r_{\text{enc}} = 30\%$. An initial learning rate of 10^{-3} is used with an exponential decay factor of 0.98 applied every 1,000 steps. We make use of positional embeddings (Appendix B.1) using $k = 16$ Gaussian random Fourier features. When performing the wall-clock training time experiment (Figure 10), we downsample the training and evaluation data to resolution 211×211.

Uncurated Reconstructions and Samples. Reconstructions of randomly selected examples from the held-out evaluation data set are shown in Figure 23. Samples from the FAE generative model and draws from the evaluation data set are shown in Figure 24.

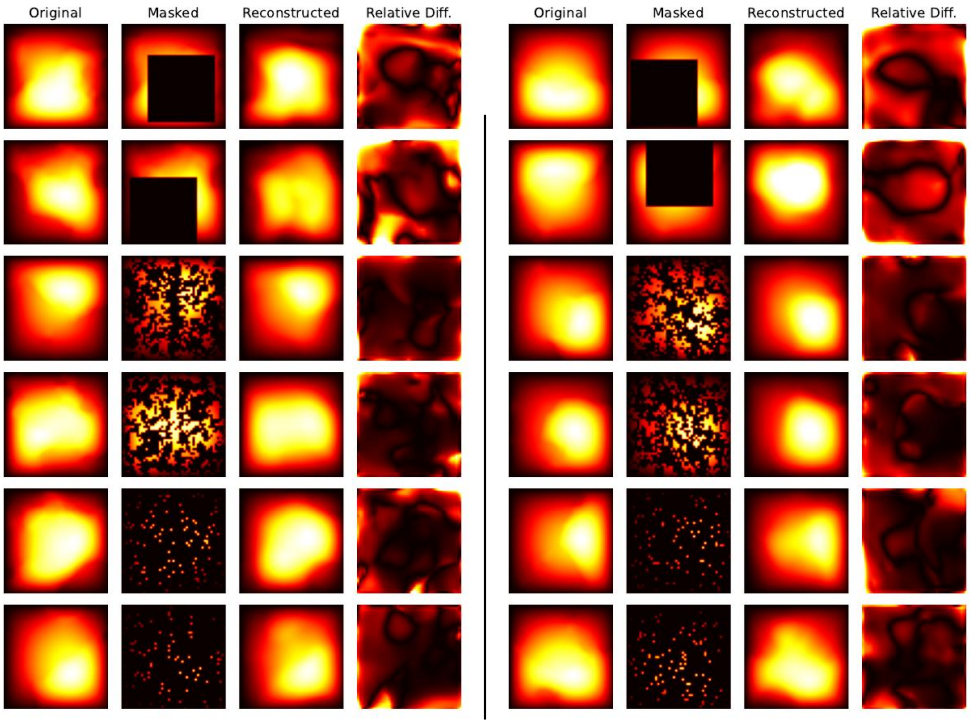
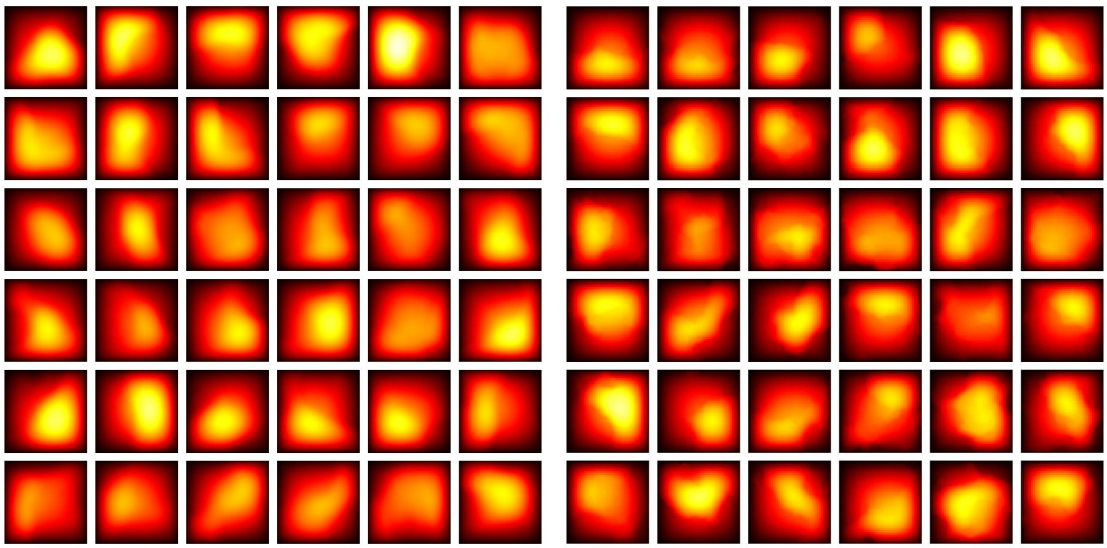


Figure 23: FAE reconstructions of Darcy flow data.



(a) FAE (b) Data set

Figure 24: Samples of Darcy flow data.

References

The Fifth International Conference on Learning Representations (ICLR 2017), 2017.

2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. IEEE.

The Ninth International Conference on Learning Representations (ICLR 2021), 2021.

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022. IEEE.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html. arXiv:1701.07875.

K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, and A. Anandkumar. Neural operators for accelerating scientific simulations and design. *Nat. Rev. Phys.*, 6: 320–328, 2024. doi:10.1038/s42254-024-00712-5.

E. Bach, R. Baptista, D. Sanz-Alonso, and A. Stuart. Inverse problems and data assimilation: A machine learning approach, 2024. arXiv:2410.10523.

J. Bengio and Y. LeCun, editors. *The Third International Conference on Learning Representations (ICLR 2015)*, 2015.

K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart. Model reduction and neural networks for parametric PDEs. *SMAI J. Comput. Math.*, 7:121–157, 2021. doi:10.5802/smai-jcm.74.

T. Bischoff and K. Deck. Unpaired downscaling of fluid flows with diffusion bridges. *Artif. Intell. Earth Syst.*, 3:e230039, 22pp., 2024. doi:10.1175/AIES-D-23-0039.1.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN 978-0-387-31073-2.

V. I. Bogachev. *Gaussian Measures*, volume 62 of *Mathematical Surveys and Monographs*. American Mathematical Society, 1998. doi:10.1090/surv/062.

K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinform.*, 22(14):e49–e57, 2006. doi:10.1093/bioinformatics/btl242.

D. R. Burt, S. W. Ober, A. Garriga-Alonso, and M. van der Wilk. Understanding variational inference in function-space. In *3rd Symposium on Advances in Approximate Bayesian Inference*, 2021. arXiv:2011.09421.

E. Calvello, N. B. Kovachki, M. E. Levine, and A. M. Stuart. Continuum attention for neural operators, 2024. arXiv:2406.06486.

G. J. Chandler and R. R. Kerswell. Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow. *J. Fluid. Mech.*, 722:554–595, 2013. doi:10.1017/jfm.2013.122.

J. T. Chang and D. Pollard. Conditioning as disintegration. *Stat. Neerl.*, 51(3):287–317, 1997. doi:10.1111/1467-9574.00056.

T. Chen and H. Chen. Approximations of continuous functionals by neural networks with application to dynamic systems. *IEEE Trans. Neural Netw.*, 4(6):910–918, 1993. doi:10.1109/72.286886.

Y. Chen, S. Liu, and X. Wang. Learning continuous image representation with local implicit image function. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* CVP (2021), pages 8624–8634. doi:10.1109/CVPR46437.2021.00852.

Y. Chen, D. Z. Huang, J. Huang, S. Reich, and A. M. Stuart. Sampling via gradient flows in the space of probability measures. *arXiv preprint arXiv:2310.03597*, 2023.

T. Cinquin and R. Bamler. Regularized KL-divergence for well-defined function-space variational inference in Bayesian neural networks, 2024. arXiv:2406.04317.

S. L. Cotter, M. Dashti, and A. M. Stuart. Approximation of Bayesian inverse problems for PDEs. *SIAM J. Numer. Anal.*, 48(1):322–345, 2010. doi:10.1137/090770734.

S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. MCMC methods for functions: Modifying old algorithms to make them faster. *Stat. Sci.*, 28(3), 2013. doi:10.1214/13-STS421.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, second edition, 2006. doi:10.1002/047174882X.

W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu. Sobolev training for neural networks. In Guyon et al. (2017), pages 4278–4287. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/758a06618c69880a6cee5314ee42d52f-Paper.pdf. arXiv:1706.04859.

M. Dashti and A. M. Stuart. The Bayesian approach to inverse problems. In *Handbook of Uncertainty Quantification. Vol. 1, 2, 3*, chapter 7, pages 311–428. Springer, Cham, 2017. doi:10.1007/978-3-319-12385-1_7.

M. V. de Hoop, D. Z. Huang, E. Qian, and A. M. Stuart. The cost-accuracy trade-off in operator learning with neural networks. *J. Mach. Learn.*, 1(3):299–341, 2022. doi:10.4208/jml.220509.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

M. M. Dunlop, D. Slepčev, A. M. Stuart, and M. Thorpe. Large data and zero noise limits of graph-based semi-supervised learning algorithms. *Appl. Comput. Harmon. Anal.*, 49 (2):655–697, 2020. doi:10.1016/j.acha.2019.03.005.

R. Durrett. *Probability: Theory and Examples*. Number 49 in Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, fifth edition, 2019. doi:10.1017/9781108591034.

W. E, W. Ren, and E. Vanden-Eijnden. Minimum action method for the study of rare events. *Comm. Pure Appl. Math.*, 57(5):637–656, 2004. doi:10.1002/cpa.20005.

H. Edwards and A. Storkey. Towards a neural statistician. In *The Fifth International Conference on Learning Representations (ICLR 2017)* ICL (2017). arXiv:1606.02185.

L. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, second edition, 2010. doi:10.1090/gsm/019.

G. Franzese, G. Corallo, S. Rossi, M. Heinonen, M. Filippone, and P. Michiardi. Continuous-time functional diffusion processes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 37370–37400. Curran Associates, Inc., 2023.

R. A. Freeze and J. A. Cherry. *Groundwater*. Prentice-Hall, Englewood Cliffs, NJ, 1979. ISBN 0-13-365312-9.

P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Schölkopf. From variational to deterministic autoencoders. In *The Eighth International Conference on Learning Representations (ICLR 2020)*, 2020. arXiv:1903.12436.

I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors. *Advances in Neural Information Processing Systems*, volume 30, 2017. Curran Associates, Inc.

P. Hagemann, L. Ruthotto, G. Steidl, and N. T. Yang. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation, 2023. arXiv:2303.04772.

M. Hairer, A. Stuart, and J. Voss. Signal processing problems on function space: Bayesian formulation, stochastic PDEs and effective MCMC methods. In *The Oxford Handbook of Nonlinear Filtering*, pages 833–873. Oxford University Press, Oxford, 2011. ISBN 9780199532902.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* CVP (2022), pages 15979–15988. doi:10.1109/CVPR52688.2022.01553.

D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs), 2016. arXiv:1606.08415.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *The Fifth International Conference on Learning Representations (ICLR 2017)* ICL (2017). URL https://openreview.net/pdf?id=Sy2fzU9gl.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In Larochelle et al. (2020), pages 6840–6851. arXiv:2006.11239.

D. Z. Huang, N. H. Nelsen, and M. Trautner. An operator learning perspective on parameter-to-observable maps. *Found. Data Sci.*, 7:163–225, 2025. doi:10.3934/fods.2024037.

B. E. Husic and V. S. Pande. Markov state models: From an art to a science. *J. Am. Chem. Soc.*, 140(7):2386–2396, 2018. doi:10.1021/jacs.7b12191.

J. B. Keller. Darcy's law for flow in porous media and the two-space method. In *Nonlinear Partial Differential Equations in Engineering and Applied Science*, pages 429–443. Routledge, 1980. doi:10.1201/9780203745465-27.

G. Kerrigan, J. Ley, and P. Smyth. Diffusion generative models in infinite dimensions. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, volume 206 of *Proceedings of Machine Learning Research*, pages 9538–9563, 2023. arXiv:2212.00886.

J. Kim, J. Yoo, J. Lee, and S. Hong. SetVAE: Learning hierarchical composition for generative modeling of set-structured data. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* CVP (2021), pages 15054–15063. doi:10.1109/CVPR46437.2021.01481.

D. P. Kingma. *Variational inference and deep learning: A new synthesis*. PhD thesis, University of Amsterdam, 2017. ISBN: 978-94-6299-745-5.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In Bengio and LeCun (2015). arXiv:1412.6980.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In Y. Bengio and Y. LeCun, editors, *The Second International Conference on Learning Representations (ICLR 2014)*, 2014. arXiv:1312.6114.

D. P. Kingma and M. Welling. An introduction to variational autoencoders. *FNT in Mach. Learn.*, 12(4):307–392, 2019. doi:10.1561/2200000056.

D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer. Machine learning–accelerated computational fluid dynamics. *Proc. Nat. Acad. Sci. USA*, 118(21): e2101784118, 8pp., 2021. doi:10.1073/pnas.2101784118.

K. A. Konovalov, I. C. Unarta, S. Cao, E. C. Goonetilleke, and X. Huang. Markov state models to study the functional dynamics of proteins in the wake of machine learning. *J. Amer. Chem. Soc. Au*, 1(9):1330–1341, 2021. doi:10.1021/jacsau.1c00254.

N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.*, 23:1–97, 2023. arXiv:2108.08481.

S. G. Krein and Yu. I. Petunin. Scales of Banach spaces. *Russ. Math. Surv.*, 21(2):85–159, 1966. doi:10.1070/RM1966v021n02ABEH004151.

S. Lanthaler, A. M. Stuart, and M. Trautner. Discretization error of Fourier neural operators, 2024. arXiv:2405.02221.

H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors. *Advances in Neural Information Processing Systems*, volume 33, 2020. Curran Associates, Inc.

S. Lee. Mesh-independent operator learning for partial differential equations. In *2nd AI4Science Workshop at the 39th International Conference on Machine Learning*, 2022. URL https://openreview.net/pdf?id=JUtZG8-2vGp.

J. Li, Z. Pei, W. Li, G. Gao, L. Wang, Y. Wang, and T. Zeng. A systematic survey of deep learning-based single-image super-resolution. *ACM Comput. Surv.*, 56(10):1–40, 2024. doi:10.1145/3659100.

Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on $\mathcal{X}$-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 820–830. Curran Associates, Inc., 2018. arXiv:1801.07791.

Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *The Ninth International Conference on Learning Representations (ICLR 2021)* ICL (2021). arXiv:2010.08895.

J. H. Lim, N. B. Kovachki, R. Baptista, C. Beckham, K. Azizzadenesheli, J. Kossaifi, V. Voleti, J. Song, K. Kreis, J. Kautz, C. Pal, A. Vahdat, and A. Anandkumar. Score-based diffusion models in function space, 2023. arXiv:2302.07400.

R. S. Liptser and A. N. Shiryaev. *Statistics of Random Processes*. Springer, Berlin, Heidelberg, second edition, 2001. doi:10.1007/978-3-662-13043-8.

L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, 2021. doi:10.1038/s42256-021-00302-5.

A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets for deep learning of molecular kinetics. *Nat. Commun.*, 9(1):5, 11pp., 2018. doi:10.1038/s41467-017-02388-1.

B. Øksendal. *Stochastic Differential Equations*. Universitext. Springer, Berlin, Heidelberg, sixth edition, 2003. doi:10.1007/978-3-642-14394-6.

B. Peherstorfer. Breaking the Kolmogorov barrier with nonlinear model reduction. *Not. Am. Math. Soc.*, 69(5):725–733, 2022. doi:10.1090/noti2475.

J. Pidstrigach, Y. Marzouk, S. Reich, and S. Wang. Infinite-dimensional diffusion models for function spaces, 2023. arXiv:2302.10130.

M. Prasthofer, T. De Ryck, and S. Mishra. Variable-input deep operator networks, 2022. arXiv:2205.11404.

J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. Markov models of molecular kinetics: Generation and validation. *J. Chem. Phys.*, 134(17):174105, 23pp., 2011. doi:10.1063/1.3565032.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85. IEEE, 2017. doi:10.1109/CVPR.2017.16.

W. Quan, J. Chen, Y. Liu, D.-M. Yan, and P. Wonka. Deep learning-based image and video inpainting: A survey. *Int. J. Comput. Vis.*, 132:2364–2400, 2024. doi:10.1007/s11263-023-01977-6.

M. A. Rahman, M. A. Florez, A. Anandkumar, Z. E. Ross, and K. Azizzadenesheli. Generative adversarial neural operators. *Transact. Mach. Learn. Res.*, 2022. arXiv:2205.03017.

J. O. Ramsay and B. W. Silverman, editors. *Applied Functional Data Analysis: Methods and Case Studies*. Springer Series in Statistics. Springer, 2002. doi:10.1007/b98886.

M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors. *Advances in Neural Information Processing Systems*, volume 34, 2021. Curran Associates, Inc.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* CVP (2022), pages 10674–10685. doi:10.1109/CVPR52688.2022.01042.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, first edition, 2019. doi:10.1017/9781108186735.

T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*, volume 21 of *Interdisciplinary Applied Mathematics*. Springer, New York, second edition, 2010. doi:10.1007/978-1-4419-6351-2.

D. W. Scott. *Multivariate Density Estimation*. Wiley Series in Probability and Statistics. John Wiley and Sons, second edition, 2015. doi:10.1002/9781118575574.

J. H. Seidman, G. Kissas, P. Perdikaris, and G. J. Pappas. NOMAD: Nonlinear manifold decoders for operator learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5601–5613. Curran Associates, Inc., 2022. arXiv:2206.03551.

J. H. Seidman, G. Kissas, G. J. Pappas, and P. Perdikaris. Variational autoencoding neural operators. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, volume 202 of *Proceedings of Machine Learning Research*, pages 30491–30522, 2023. arXiv:2302.10351.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Bengio and LeCun (2015). arXiv:1409.1556.

V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In Larochelle et al. (2020), pages 7462–7473. arXiv:2006.09661.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, 2015. arXiv:1503.03585.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth International Conference on Learning Representations (ICLR 2021)* ICL (2021). arXiv:2011.13456.

A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numer.*, 19:451–559, 2010. doi:10.1017/S0962492910000061.

V. N. Sudakov. Linear sets with quasi-invariant measure. *Dokl. Akad. Nauk SSSR*, 127: 524–525, 1959.

T. J. Sullivan. *Introduction to Uncertainty Quantification*, volume 63 of *Texts in Applied Mathematics*. Springer, 2015. doi:10.1007/978-3-319-23395-6.

S. Sun, G. Zhang, J. Shi, and R. Grosse. Functional variational Bayesian neural networks. In *The Seventh International Conference on Learning Representations (ICLR 2019)*, 2019. arXiv:1903.05779.

M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In Larochelle et al. (2020), pages 7537–7547. arXiv:2006.10739.

A. Vahdat, K. Kreis, and J. Kautz. Score-based generative modeling in latent space. In Ranzato et al. (2021), pages 11287–11302. arXiv:2106.05931.

Y. Wang, D. Blei, and J. P. Cunningham. Posterior collapse and latent variable non-identifiability. In Ranzato et al. (2021), pages 5443–5455. arXiv:2301.00537.

M. Yang, B. Dai, H. Dai, and D. Schuurmans. Energy-based processes for exchangeable data. In H. Daumé III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pages 10681–10692, 2020. arXiv:2003.07521.

M. Zaheer, S. Kottur, S. Ravanbhakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In Guyon et al. (2017), pages 3391–3401. arXiv:1703.06114.

B. Zhang and P. Wonka. Functional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4723–4732, 2024. arXiv:2311.15435.