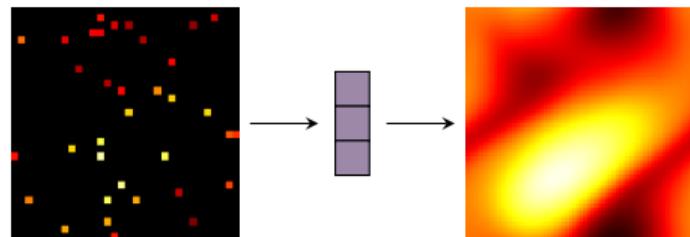


Autoencoders in Function Space

Justin Bunker¹, Mark Girolami^{1,3}, Hefin Lambley⁴,
Andrew M. Stuart², and T. J. Sullivan⁴



1



2

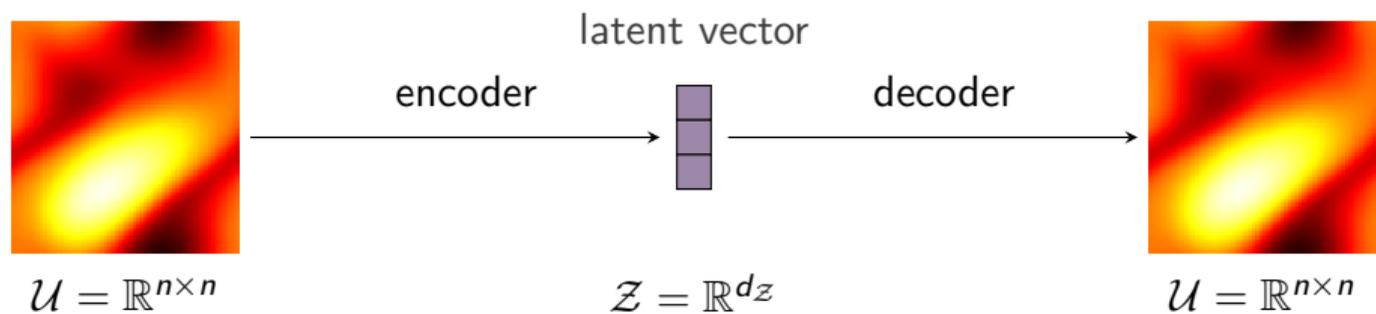


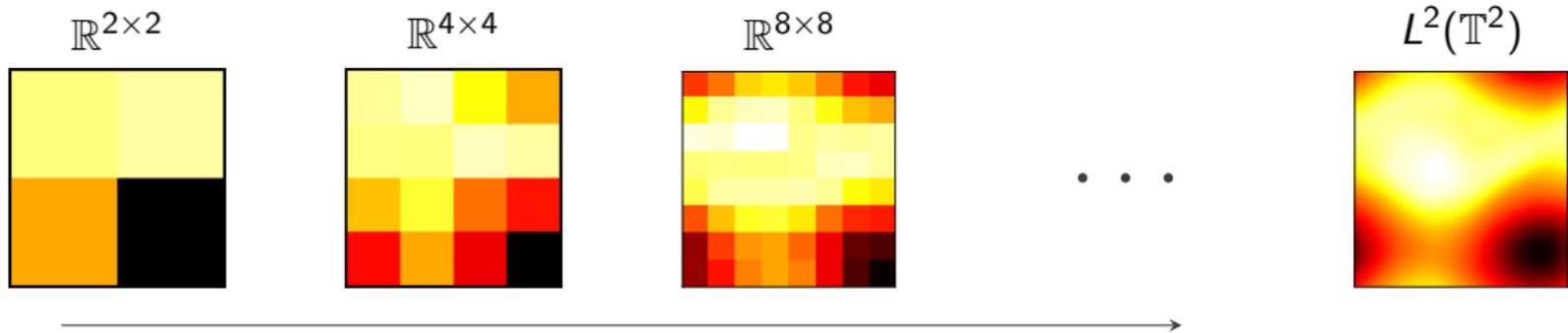
3



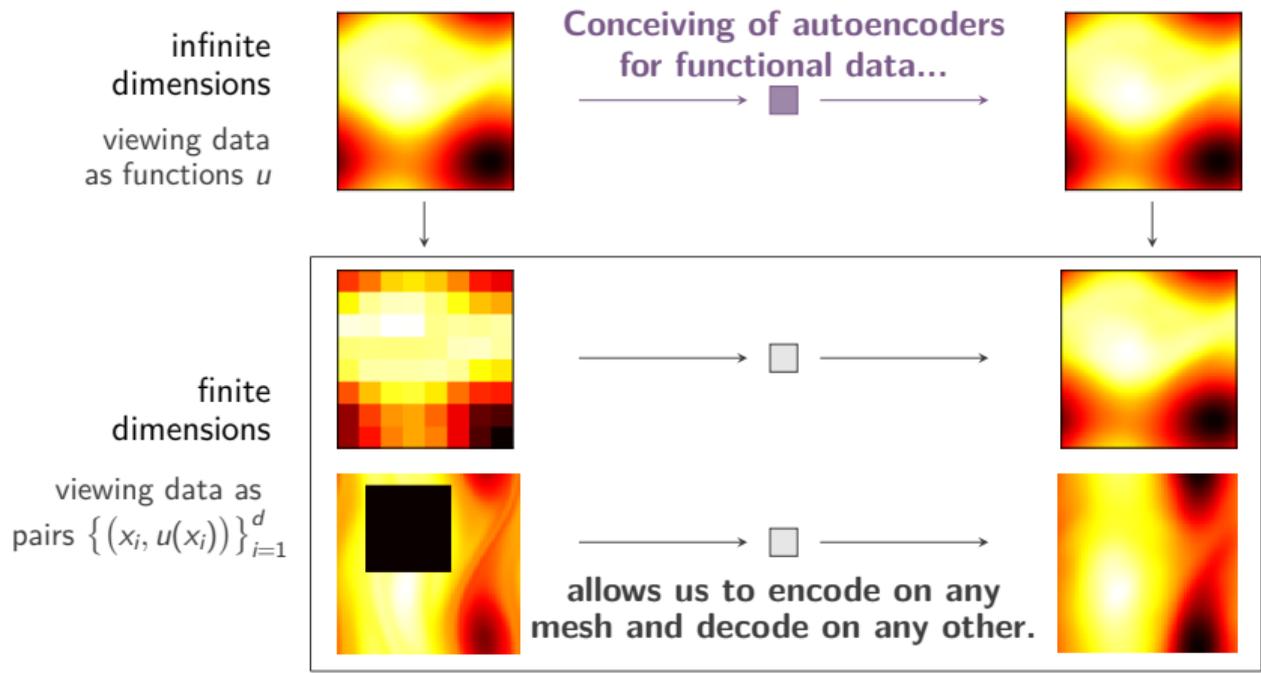
4

Autoencoders are machine-learning models for **dimension reduction** and **generative modelling**





In **scientific applications** and in **image processing**, it is useful to **view discretised data as approximations of the underlying functions**.



Given

Learn

Choose

\mathcal{U}

data space
(separable Banach space)

encoder

decoder

$\mathcal{Z} = \mathbb{R}^{d_z}$

latent space

$\{u_i\}_{i=1}^N \sim \Upsilon$

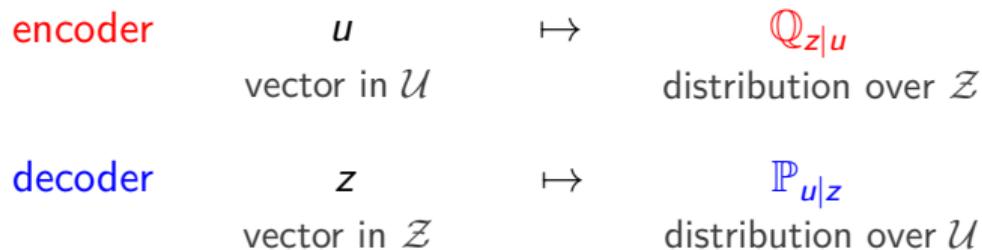
samples from
data distribution

\mathbb{P}_z

latent distribution

Functional variational autoencoder (FVAE)

Idea: view the encoder and decoder as probabilistic.



Choose the following:

family of encoders

$$\left(u \mapsto Q_{z|u}^\theta \right)_{\theta \in \Theta}$$

family of decoders

$$\left(z \mapsto P_{u|z}^\psi \right)_{\psi \in \Psi}$$

latent distribution

$$P_z \text{ on } \mathcal{Z}$$

Choose the following:

family of encoders

family of decoders

latent distribution

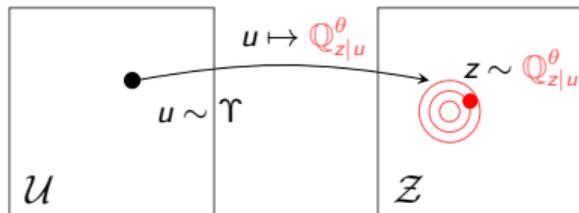
$$\left(u \mapsto Q_{z|u}^\theta \right)_{\theta \in \Theta}$$

$$\left(z \mapsto P_{u|z}^\psi \right)_{\psi \in \Psi}$$

$$P_z \text{ on } \mathcal{Z}$$

joint encoder model $Q_{z,u}^\theta$

$$Q_{z,u}^\theta(dz, du) = \Upsilon(du) Q_{z|u}^\theta(dz)$$



Choose the following:

family of encoders

family of decoders

latent distribution

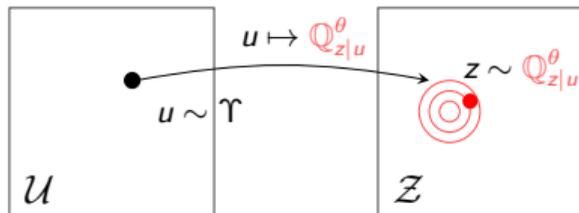
$$\left(u \mapsto Q_{z|u}^\theta \right)_{\theta \in \Theta}$$

$$\left(z \mapsto P_{u|z}^\psi \right)_{\psi \in \Psi}$$

$$P_z \text{ on } \mathcal{Z}$$

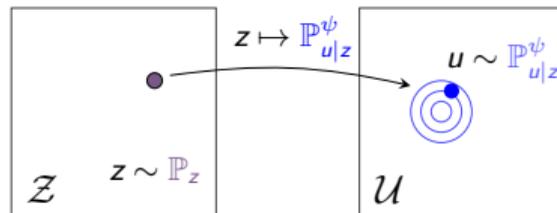
joint encoder model $Q_{z,u}^\theta$

$$Q_{z,u}^\theta(dz, du) = \Upsilon(du) Q_{z|u}^\theta(dz)$$



joint decoder model $P_{z,u}^\psi$

$$P_{z,u}^\psi(dz, du) = P_z(dz) P_{u|z}^\psi(du)$$



Choose the following:

family of encoders

family of decoders

latent distribution

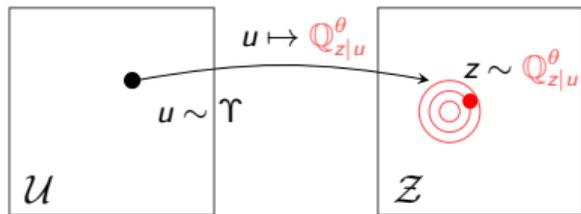
$$\left(u \mapsto Q_{z|u}^\theta \right)_{\theta \in \Theta}$$

$$\left(z \mapsto P_{u|z}^\psi \right)_{\psi \in \Psi}$$

$$P_z \text{ on } \mathcal{Z}$$

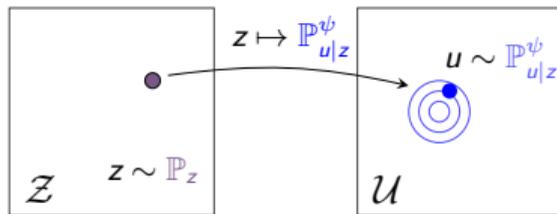
joint encoder model $Q_{z,u}^\theta$

$$Q_{z,u}^\theta(dz, du) = \Upsilon(du) Q_{z|u}^\theta(dz)$$



joint decoder model $P_{z,u}^\psi$

$$P_{z,u}^\psi(dz, du) = P_z(dz) P_{u|z}^\psi(du)$$



Objective Minimise the Kullback–Leibler divergence D_{KL} between joint distributions:

$$\arg \min_{\theta \in \Theta, \psi \in \Psi} D_{\text{KL}}(Q_{z,u}^\theta \parallel P_{z,u}^\psi).$$

When is the FVAE objective valid?

Adopt the standard Gaussian VAE model:

Gaussian encoder family $Q_{z|u}^\theta : u \mapsto N(f(u; \theta), \Sigma(u; \theta))$

Gaussian decoder family $P_{u|z}^\psi : z \mapsto N(g(z; \theta), \beta I_U)$

Gaussian latent distribution $P_z = N(0, I_Z)$

When is the FVAE objective valid?

Adopt the standard Gaussian VAE model:

Gaussian encoder family $Q_{z|u}^\theta : u \mapsto N(f(u; \theta), \Sigma(u; \theta))$

Gaussian decoder family $P_{u|z}^\psi : z \mapsto N(g(z; \theta), \beta h_u)$

Gaussian latent distribution $P_z = N(0, I_z)$

Finite dimensions

$$\mathcal{U} = \mathbb{R}^d$$

Υ has 'nice' density ν

FVAE is equivalent to a VAE:

$$\longrightarrow D_{\text{KL}}(Q_{z,u}^\theta \| P_{z,u}^\psi) = \text{usual VAE objective} + \text{finite const.}$$

evidence lower bound
(ELBO)

When is the FVAE objective valid?

Adopt the standard Gaussian VAE model:

Gaussian encoder family $Q_{z|u}^\theta: u \mapsto N(f(u; \theta), \Sigma(u; \theta))$

Gaussian decoder family $P_{u|z}^\psi: z \mapsto N(g(z; \theta), \beta h_u)$

Gaussian latent distribution $P_z = N(0, I_z)$

Finite dimensions

$$\mathcal{U} = \mathbb{R}^d$$

Υ has 'nice' density ν

FVAE is equivalent to a VAE:

$$\longrightarrow D_{\text{KL}}(Q_{z,u}^\theta \| P_{z,u}^\psi) = \text{usual VAE objective} + \text{finite const.}$$

evidence lower bound (ELBO)

Infinite dimensions

$$\mathcal{U} = L^2(0, 1)$$

Υ is *any* probability distribution on \mathcal{U} .

FVAE's objective is identically infinite:

$$\longrightarrow D_{\text{KL}}(Q_{z,u}^\theta \| P_{z,u}^\psi) = +\infty \text{ for all parameters } \theta \text{ and } \psi.$$

What went wrong?

1. $D_{\text{KL}}(\mathbb{Q}_{z,u}^\theta \parallel \mathbb{P}_{z,u}^\psi)$ is **finite** only if $\mathbb{Q}_{z,u}^\theta$ is **absolutely continuous** wrt. $\mathbb{P}_{z,u}^\psi$,
i.e., if $\mathbb{P}_{z,u}^\psi(A) = 0$, then $\mathbb{Q}_{z,u}^\theta(A) = 0$.
2. Using **Gaussian white noise** in the decoder causes a lack of absolute continuity:
 $\mathbb{P}_{u|z}^\psi = g(z; \psi) + N(0, \beta I_U)$ and realisations of $N(0, \beta I_U)$ are almost surely not L^2 .

What went wrong?

1. $D_{\text{KL}}(\mathbb{Q}_{z,u}^\theta \parallel \mathbb{P}_{z,u}^\psi)$ is **finite** only if $\mathbb{Q}_{z,u}^\theta$ is **absolutely continuous** wrt. $\mathbb{P}_{z,u}^\psi$, i.e., if $\mathbb{P}_{z,u}^\psi(A) = 0$, then $\mathbb{Q}_{z,u}^\theta(A) = 0$.
2. Using **Gaussian white noise** in the decoder causes a lack of absolute continuity: $\mathbb{P}_{u|z}^\psi = g(z; \psi) + N(0, \beta I_U)$ and realisations of $N(0, \beta I_U)$ are almost surely not L^2 .

Precisely: $D_{\text{KL}}(\mathbb{Q}_{z,u}^\theta \parallel \mathbb{P}_{z,u}^\psi) = +\infty$ as $\mathbb{Q}_{z,u}^\theta$ is not absolutely continuous wrt. $\mathbb{P}_{z,u}^\psi$:

$$\mathbb{Q}_{z,u}^\theta(\mathcal{Z} \times \mathcal{U}) = \Upsilon(u) \mathbb{Q}_{z|u}^\theta(\mathcal{Z}) = 1 \times 1 = 1$$

$$\mathbb{P}_{z,u}^\psi(\mathcal{Z} \times \mathcal{U}) = \mathbb{P}_{u|z}^\psi(\mathcal{U}) \mathbb{P}_z(\mathcal{Z}) = 0 \times 1 = 0.$$

For the FVAE objective to be valid, we must choose the data and decoder to be compatible

Assume \mathcal{U} is a separable Banach space, and take

Gaussian encoder family $\mathbb{Q}_{z|u}^\theta : u \mapsto \mathcal{N}(f(u; \theta), \Sigma(u; \theta))$

Noise distribution \mathbb{P}_η on \mathcal{U}

Shifted decoder family $\mathbb{P}_{u|z}^\psi : z \mapsto g(z; \theta) + \mathbb{P}_\eta$

Gaussian latent distribution $\mathbb{P}_z = \mathcal{N}(0, I_Z)$

Theorem

If $D_{\text{KL}}(\Upsilon \parallel \mathbb{P}_\eta) < \infty$, then the objective is **well defined**:

$$\inf_{\theta \in \Theta, \psi \in \Psi} D_{\text{KL}}(\mathbb{Q}_{z,u}^\theta \parallel \mathbb{P}_{z,u}^\psi) < \infty.$$

Examples where FVAE can and cannot be applied

- ✓ Υ is path distribution of SDE $du_t = b(u_t) dt + \sqrt{\varepsilon} dw_t$, $t \in [0, T]$;
 \mathbb{P}_η is scaled Brownian motion $d\eta_t = \sqrt{\varepsilon} dw_t$.
- ✓ Υ is posterior distribution over function (e.g., from Bayesian inverse problem);
 \mathbb{P}_η is Gaussian prior distribution.
- × Υ is distribution of natural images, viewed as functions (e.g., faces);
very hard to choose \mathbb{P}_η such that $D_{\text{KL}}(\Upsilon \parallel \mathbb{P}_\eta) < \infty$.

Example: FVAE for stochastic differential equations

On $\mathcal{U} = C([0, T], \mathbb{R}^m)$:

fix Υ distribution of $du_t = b(u_t) dt + \sqrt{\varepsilon} dw_t, \quad u_0 = 0, \quad t \in [0, T]$
choose \mathbb{P}_η distribution of $d\eta_t = \sqrt{\varepsilon} dw_t, \quad \eta_0 = 0, \quad t \in [0, T].$

Example: FVAE for stochastic differential equations

On $\mathcal{U} = C([0, T], \mathbb{R}^m)$:

fix Υ distribution of $du_t = b(u_t) dt + \sqrt{\varepsilon} dw_t, \quad u_0 = 0, \quad t \in [0, T]$

choose \mathbb{P}_η distribution of $d\eta_t = \sqrt{\varepsilon} dw_t, \quad \eta_0 = 0, \quad t \in [0, T].$

Proposition: finite infimum of FVAE objective

By the Girsanov theorem, assuming b is “nice”,

$$D_{\text{KL}}(\Upsilon \parallel \mathbb{P}_\eta) = \mathbb{E}_{u \sim \Upsilon} \left[\frac{1}{2\varepsilon} \int_0^T \|b(u_t)\|^2 dt \right].$$

Example: FVAE for stochastic differential equations

On $\mathcal{U} = C([0, T], \mathbb{R}^m)$:

fix Υ distribution of $du_t = b(u_t) dt + \sqrt{\varepsilon} dw_t, \quad u_0 = 0, \quad t \in [0, T]$

choose \mathbb{P}_η distribution of $d\eta_t = \sqrt{\varepsilon} dw_t, \quad \eta_0 = 0, \quad t \in [0, T].$

Proposition: finite infimum of FVAE objective

By the Girsanov theorem, assuming b is “nice”,

$$D_{\text{KL}}(\Upsilon \parallel \mathbb{P}_\eta) = \mathbb{E}_{u \sim \Upsilon} \left[\frac{1}{2\varepsilon} \int_0^T \|b(u_t)\|^2 dt \right].$$

Theorem: FVAE objective for stochastic differential equations

$$D_{\text{KL}}(\mathbb{Q}_{z,u}^\theta \parallel \mathbb{P}_{z,u}^\psi) = \mathbb{E}_{u \sim \Upsilon} [\mathcal{L}(u; \theta, \psi)] + D_{\text{KL}}(\Upsilon \parallel \mathbb{P}_\eta),$$

$$\mathcal{L}(u; \theta, \psi) = \mathbb{E}_{z \sim \mathbb{Q}_{z|u}^\theta} \left[\frac{1}{\varepsilon} \langle g(z; \psi), u \rangle_{H^1} - \frac{1}{2\varepsilon} \|g(z; \psi)\|_{H^1}^2 \right] + D_{\text{KL}}(\mathbb{Q}_{z|u}^\theta \parallel \mathbb{P}_z)$$

↪ Similar arguments apply to other noise processes, e.g., OU noise.

Our proposed mesh-invariant encoder architecture

In function space Define MLPs κ and ρ and let

$$f(u; \theta) = \rho \left(\int_{\Omega} \kappa(x, u(x); \theta) dx; \theta \right).$$

We use a similar parametrisation for $\Sigma(u; \theta)$.

For discrete data Represent discrete functions as point clouds $(u(x_k))_{k=1}^N$ on mesh $(x_k)_{k=1}^N$.
Then, approximate integral by sum:

$$\left((x_k)_{k=1}^N, (u(x_k))_{k=1}^N \right) \mapsto \rho \left(\frac{1}{N} \sum_{k=1}^N \kappa(x_k, u(x_k); \theta); \theta \right)$$

Our proposed mesh-invariant decoder architecture

In function space Parametrise g as a coordinate MLP

$$g(z; \psi)(x) = \gamma(z, x; \psi).$$

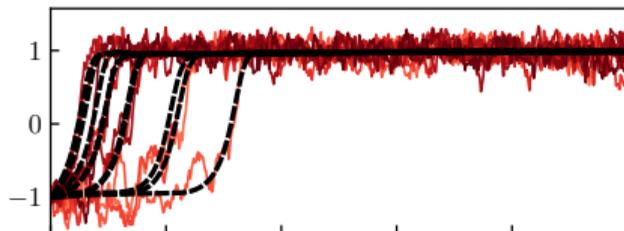
For discrete data On output mesh $(y_k)_{k=1}^M$, represent by point cloud

$$z \mapsto \left((y_k)_{k=1}^M, (\gamma(z, y_k; \psi))_{k=1}^M \right)$$

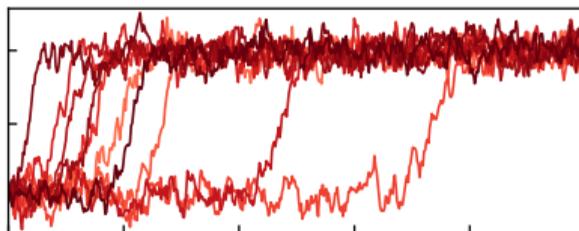
Example problem: Brownian dynamics

Data: Υ distribution on $\mathcal{U} = C([0, 5], \mathbb{R})$ of $du_t = -\nabla U(u_t) dt + \sqrt{\varepsilon} dw_t$, $u_0 = -1$,

(a) Data $u \sim \Upsilon$ and reconstructions $g(f(u))$



(b) Samples from generative model



More applications of FVAE in our paper, e.g., motivated by **molecular dynamics** learning a Markov state model from **irregularly sampled transition paths**.

Functional autoencoder (FAE)

Idea: view the encoder and decoder as deterministic.

$$\text{encoder} \quad \mathcal{U} \ni u \quad \mapsto \quad f(u) \in \mathcal{Z}$$

$$\text{decoder} \quad \mathcal{Z} \ni z \quad \mapsto \quad g(z) \in \mathcal{U}$$

Then choose:

$$\left(u \mapsto f(u; \theta) \right)_{\theta \in \Theta} \quad \text{family of encoders}$$

$$\left(z \mapsto g(z; \psi) \right)_{\psi \in \Psi} \quad \text{family of decoders}$$

Objective: Given **regularisation scale** $\beta > 0$, solve

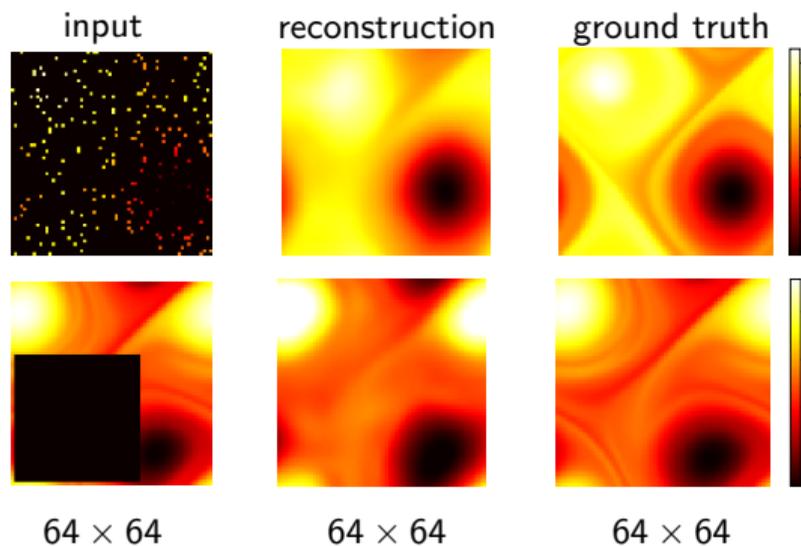
$$\arg \min_{\theta \in \Theta, \psi \in \Psi} \mathbb{E}_{u \sim \Upsilon} \left[\frac{1}{2} \|g(f(u; \theta); \psi) - u\|_{\mathcal{U}}^2 + \beta \|f(u; \theta)\|_2^2 \right].$$

\rightsquigarrow Similar to the VAE objective in finite dimensions with Gaussian model.

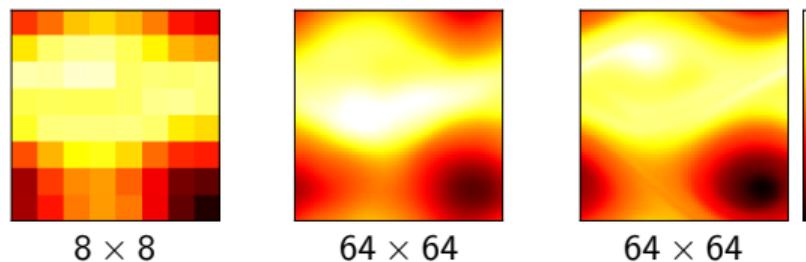
✓ Objective has finite infimum as long as $\mathbb{E}_{u \sim \Upsilon} [\|u\|^2] < \infty$

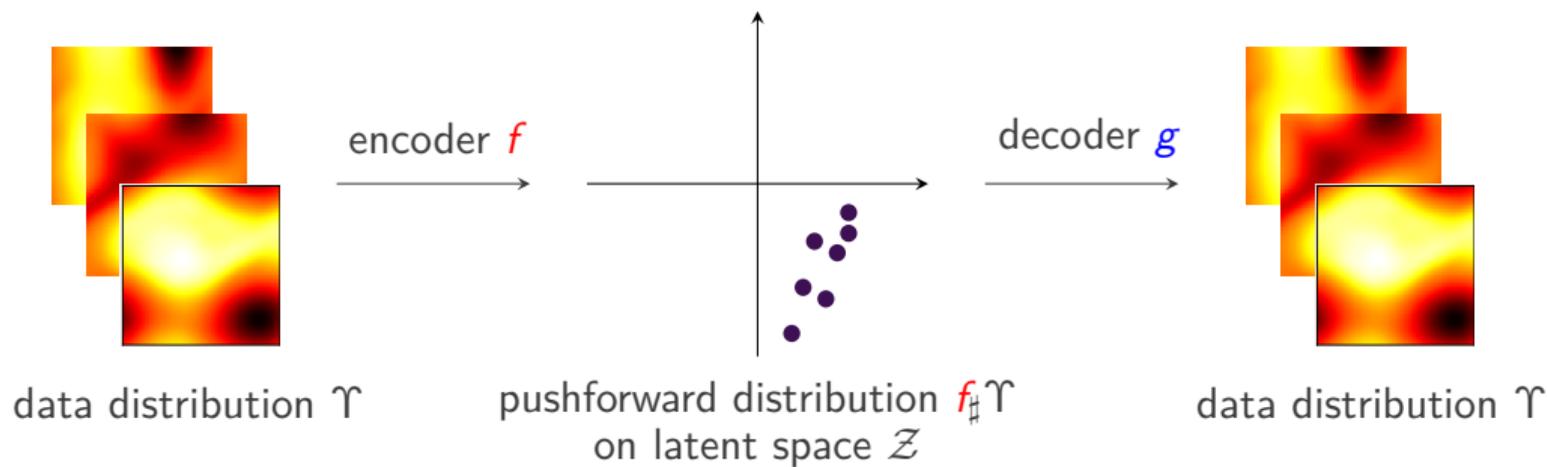
New applications to **inpainting** and **superresolution**

Inpainting
trained at 64×64



Data-driven superresolution
trained at 64×64





Latent generative models While FAE is not inherently a generative model, can learn generative model $\mathbb{P}_{\mathcal{Z}}$ to approximate $f_{\#}\Upsilon$ on \mathcal{Z} similar to image generative models such as Stable Diffusion.

Summing things up...

- **Functional variational autoencoder (FVAE)**
Probabilistic generative model with **built-in uncertainty quantification**.
Works for **specific classes of data distributions**.
- **Functional autoencoder (FAE)**
Non-probabilistic autoencoder that can be augmented with a generative model
Works for **most data distributions** on function space.

Limitations and future work

1. **Barriers to variational inference** in function space;
can VAEs be extended without the stringent constraints of FVAE?
2. Need for **better architectures** that can be evaluated on any mesh
e.g., point-cloud architectures such as PointCNN.
3. FVAE and FAE could serve as **building block** for
 - supervised learning \rightsquigarrow inspired by PCA-NET
 - generative modelling \rightsquigarrow inspired by Stable Diffusion.

More details in our paper:

Justin Bunker, Mark Girolami, **Hefin Lambley**, Andrew M. Stuart, and T. J. Sullivan.
Autoencoders in Function Space. JMLR **26**(165):1–54.

Code package in Python + JAX available at:

https://github.com/htlambley/functional_autoencoders