Pulling back information geometry

Miguel González-Duque, Geometric DL Reading Group, 2023

About me

B.Sc. and M.Sc. in Mathematics

Ph.D Fellow at the IT
University of Copenhagen

Working on:

Applications of deep generative
models to video games

About today

Pulling back information geometry





LATENT SPACE ODDITY: ON THE CURVATURE OF DEEP GENERATIVE MODELS

Georgios Arvanitidis, Lars Kai Hansen, Søren Hauberg
Technical University of Denmark, Section for Cognitive Systems
{gear,lkai,sohau}@dtu.dk

Learning Riemannian Manifolds for Geodesic Motion Skills

Hadi Beik-Mohammadi[1,2], Søren Hauberg[3], Georgios Arvanitidis[4], Gerhard Neumann[2], and Leonel Rozo[1]

Article | Open Access | Published: 08 April 2022

Learning meaningful representations of protein sequences

Nicki Skafte Detlefsen, Søren Hauberg & Wouter Boomsma

About today

Pulling back information geometry





LATENT SPACE ODDITY: ON THE CURVATURE OF DEEP GENERATIVE MODELS

Learning Riemannian Manifolds for

The set-up (Variational Autoencoders & geometry)

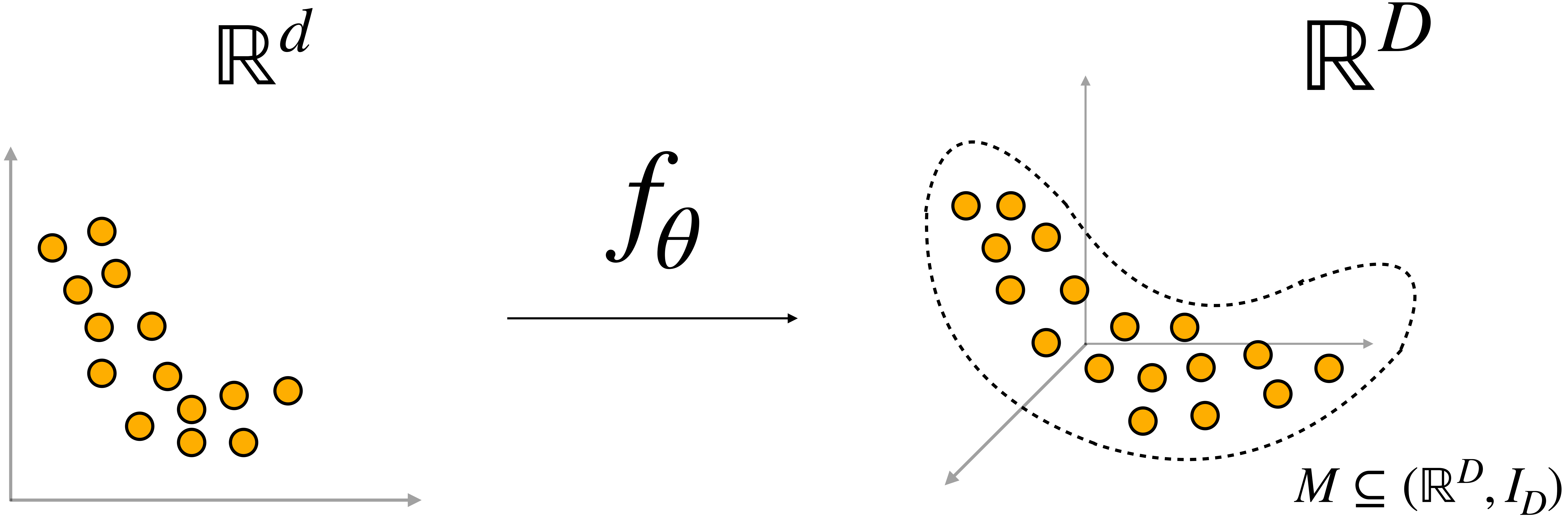
Nicki Skafte Detlefsen, Søren Hauberg & Wouter Boomsma

Applications of latent space geometry

Pulling back information geometry

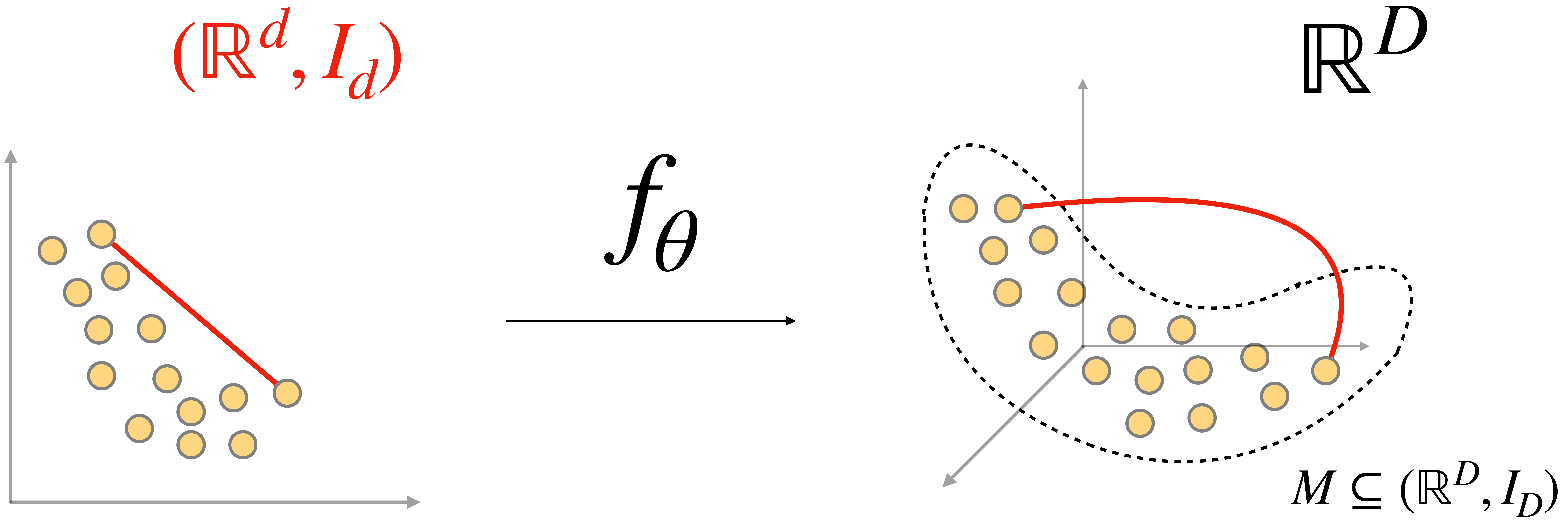
Questions: anytime

The set-up



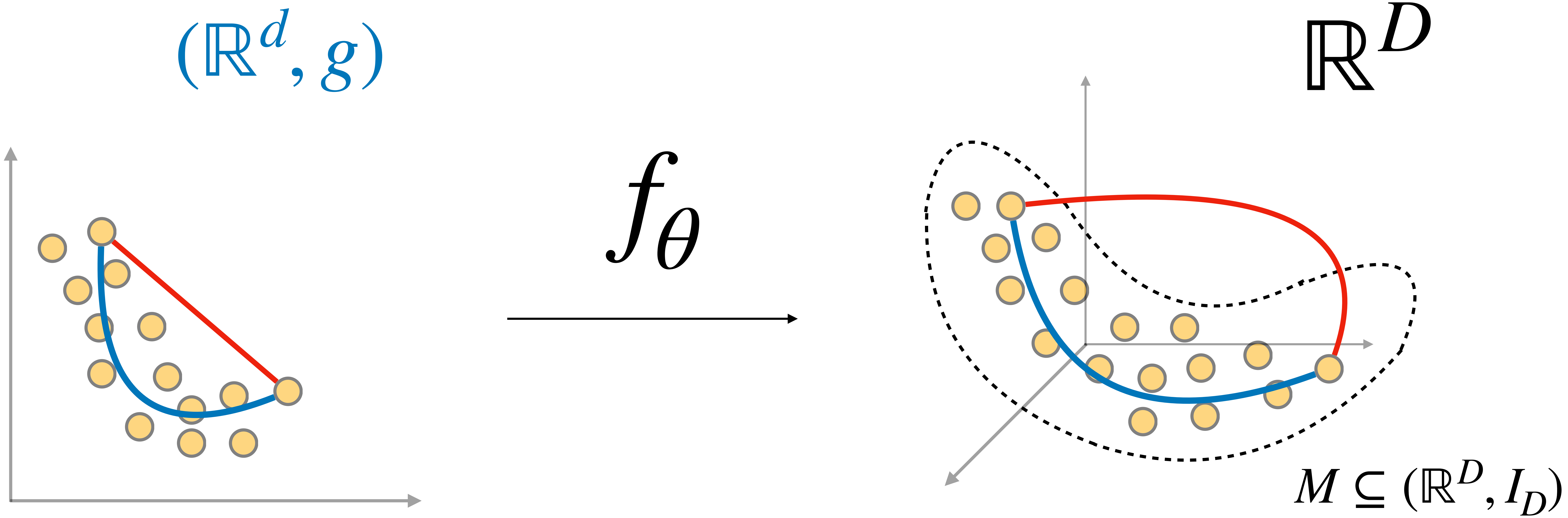
The set-up

Learning latent representations



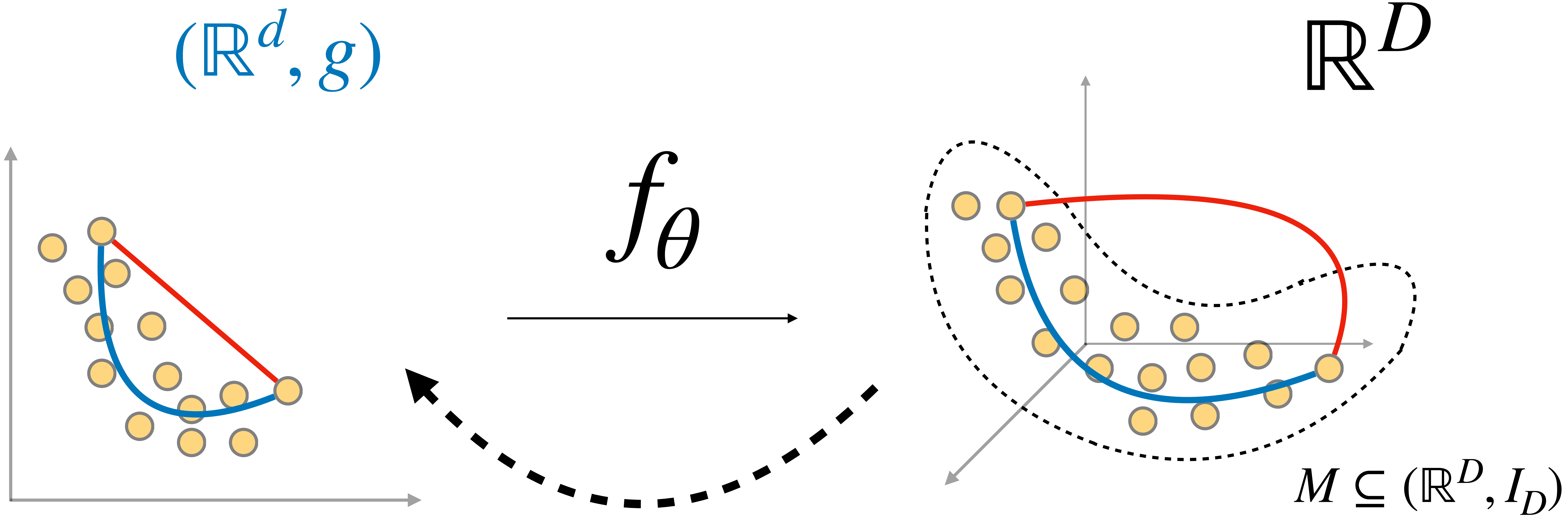
The set-up

Learning latent representations



The set-up

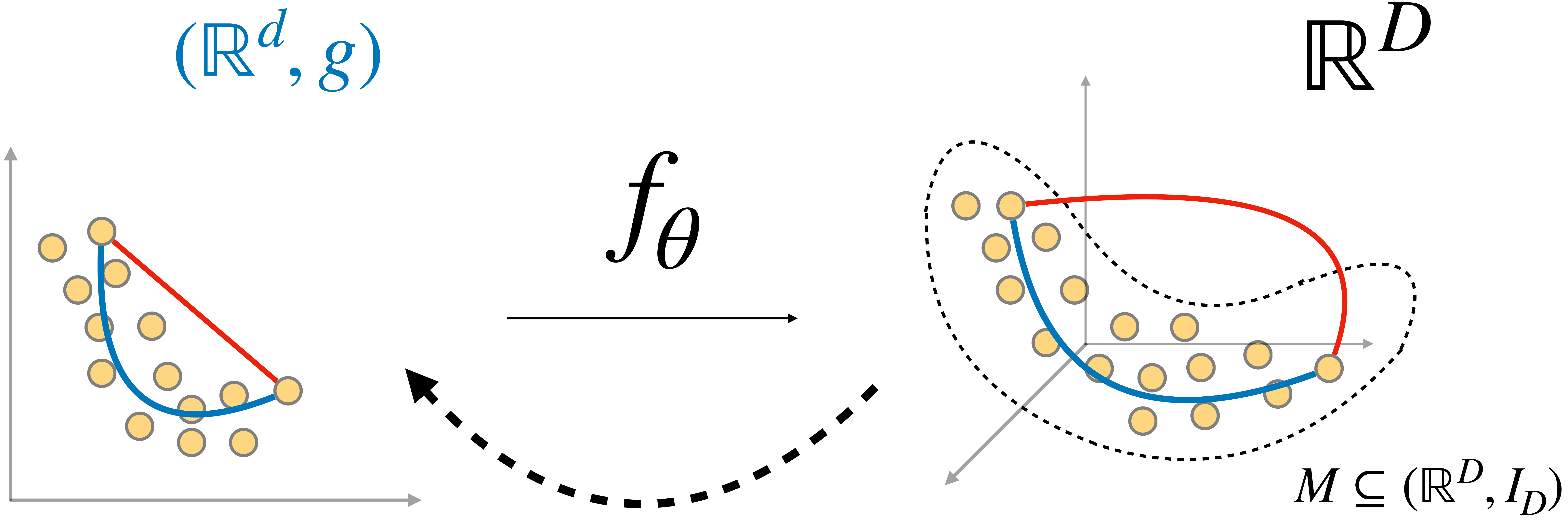
Learning latent representations



$$[g(z)] = J_{f_\theta}(z)^\top J_{f_\theta}(z)$$

The set-up

Learning latent representations

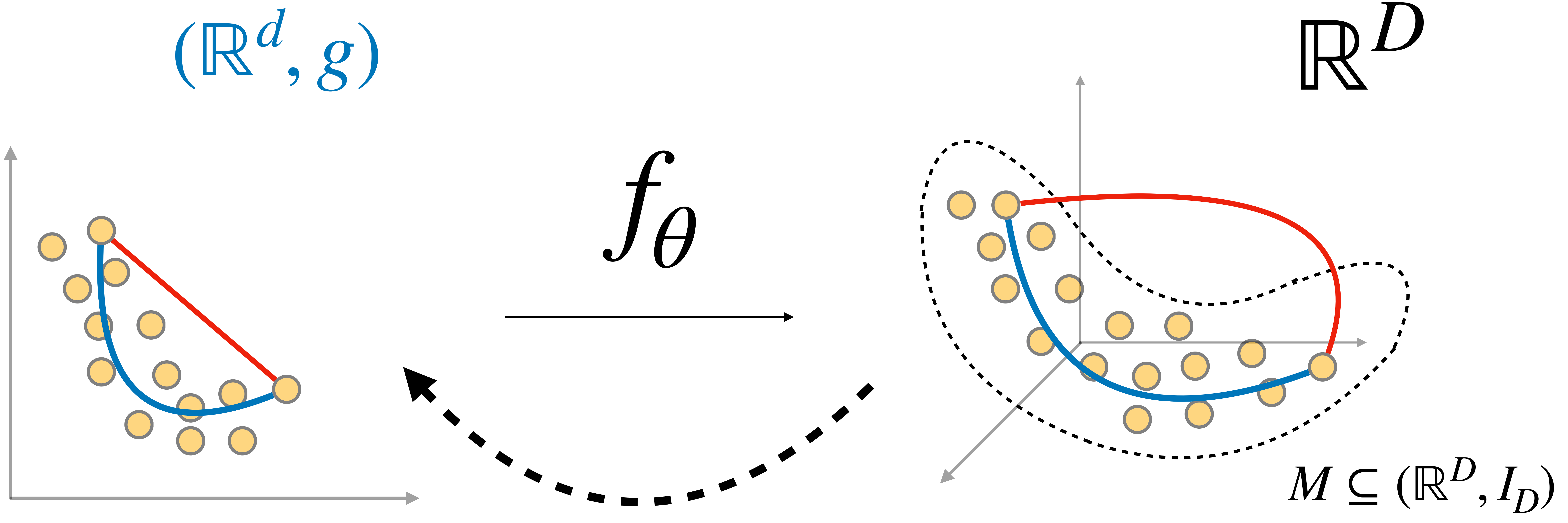


$$[g(z)] = J_{f_\theta}(z)^\top J_{f_\theta}(z)$$

The *pullback metric*.

he set-up

Learning latent representations



$$[g(z)] = J_{f_\theta}(z)^\top J_{f_\theta}(z)$$

The *pullback metric*.

In our set-up: f_θ is the decoder of a VAE.

The *pullback metric*.

Informally

Given a curve $c\colon [0,1] \to \mathscr{Z}$ and an immersion $f_\theta\colon \mathscr{Z} \to \mathbb{R}^D$...

$$
\begin{aligned}
\text{Length}[c] &= \int_0^1 \|(f_\theta \circ c)'(t)\| \mathrm{d}t \\
&= \dots \\
&= \int_0^1 \sqrt{c'(t)^\top J_{f_\theta}(c(t))^\top J_{f_\theta}(c(t)) c'(t)}
\end{aligned}
$$

The *pullback metric*.

Informally

Given a curve $c\colon [0,1] \to \mathscr{Z}$ and an immersion $f_\theta\colon \mathscr{Z} \to \mathbb{R}^D$...

$$
\begin{aligned}
\text{Length}[c] &= \int_0^1 \|(f_\theta \circ c)'(t)\| \mathrm{d}t \\
&= \dots \\
&= \int_0^1 \sqrt{c'(t)^\top J_{f_\theta}(c(t))^\top J_{f_\theta}(c(t)) c'(t)}
\end{aligned}
$$

Curves that locally minimize length: geodesics.

The *pullback metric*.

Informally

Given a curve $c: [0,1] \to \mathscr{Z}$ and an immersion $f_\theta: \mathscr{Z} \to \mathbb{R}^D$…

$$\text{Length}[c] = \int_0^1 \|(f_\theta \circ c)'(t)\| \mathrm{d}t$$
$$= \ldots$$
$$= \int_0^1 \sqrt{c'(t)^\top J_{f_\theta}(c(t))^\top J_{f_\theta}(c(t)) c'(t)}$$

Curves that locally minimize length: geodesics.

Formally

Def. A metric g takes two tangent vectors and computes their inner product.

Given an immersion $f_\theta: \mathscr{Z} \to (M, g)$, we define a metric on $\mathscr{Z}$ given by

$$g_{f(z)}((\mathrm{d}f)_z(v_1), (\mathrm{d}f)_z(v_2))$$

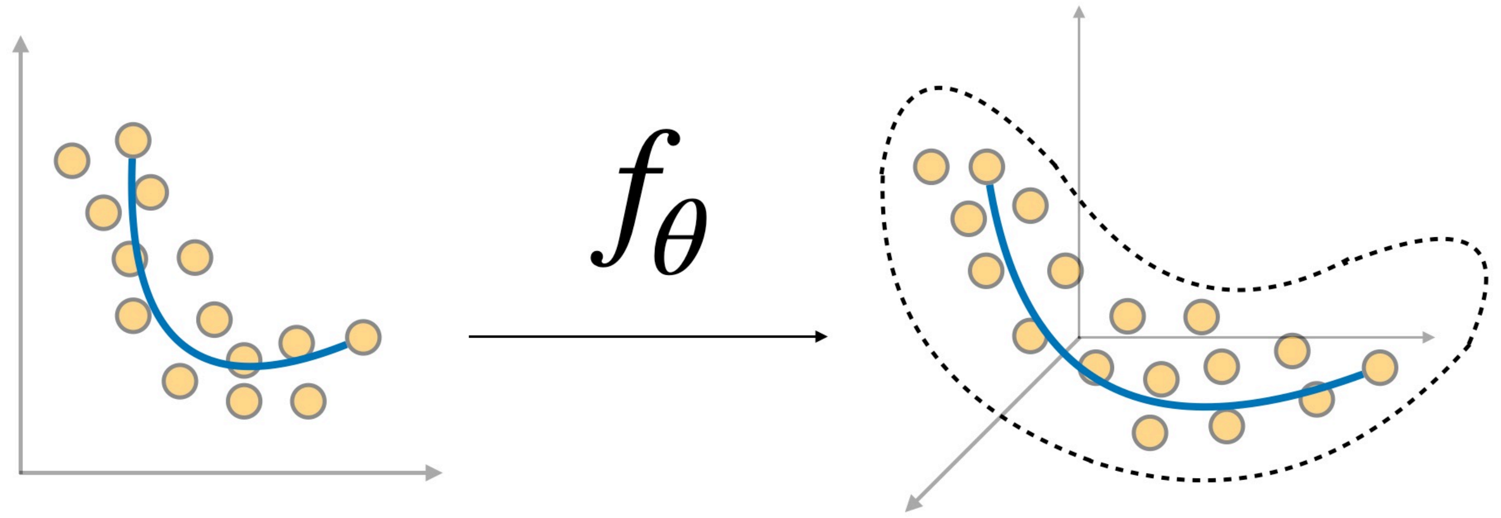
The *pullback metric*.

Pro:

Distances are defined in data space
(i.e. invariant to reparametrizations)

Con:

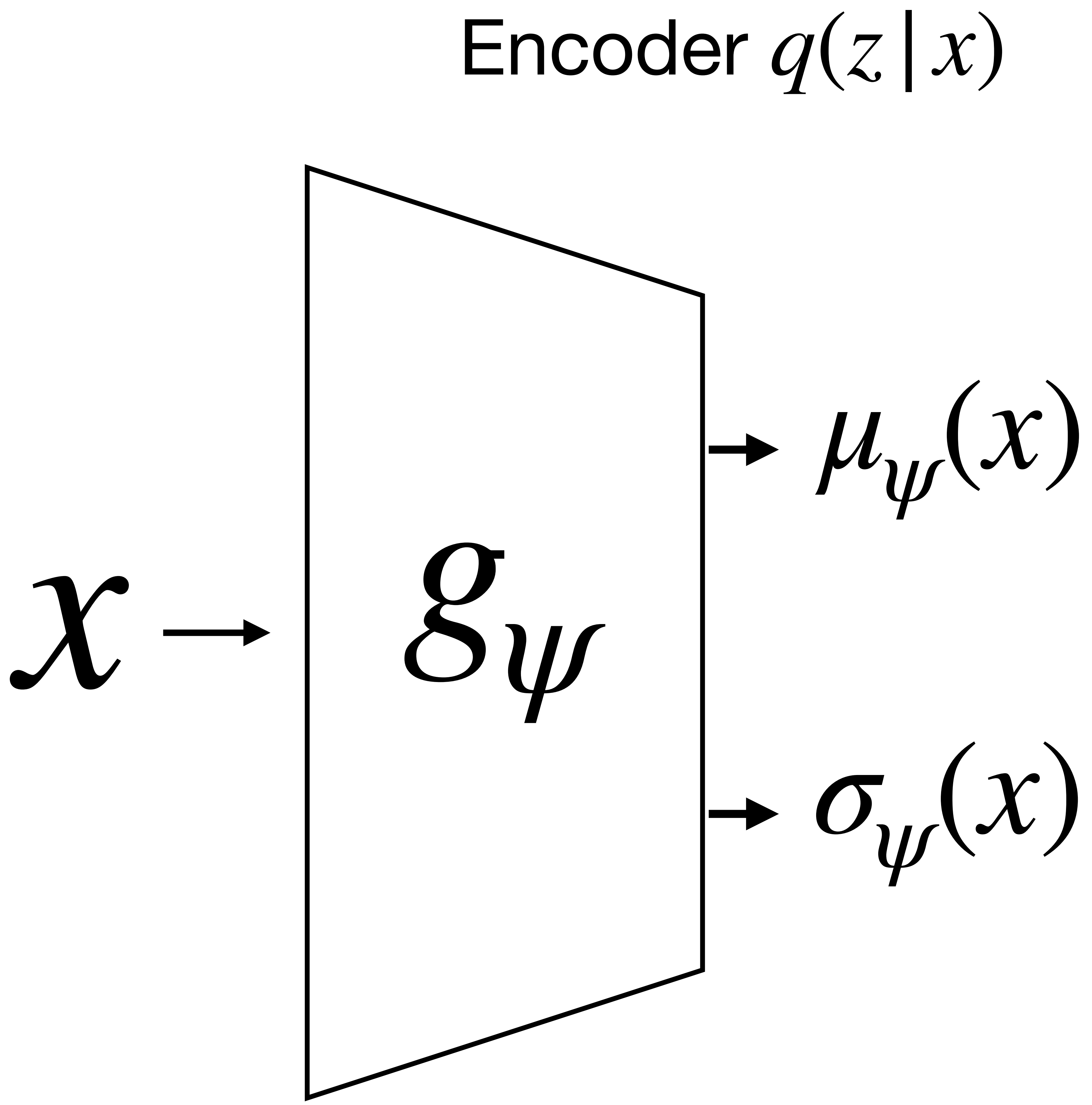
We have to compute $J_{f_\theta}(z)^\top J_{f_\theta}(z)$
for a given likelihood



x

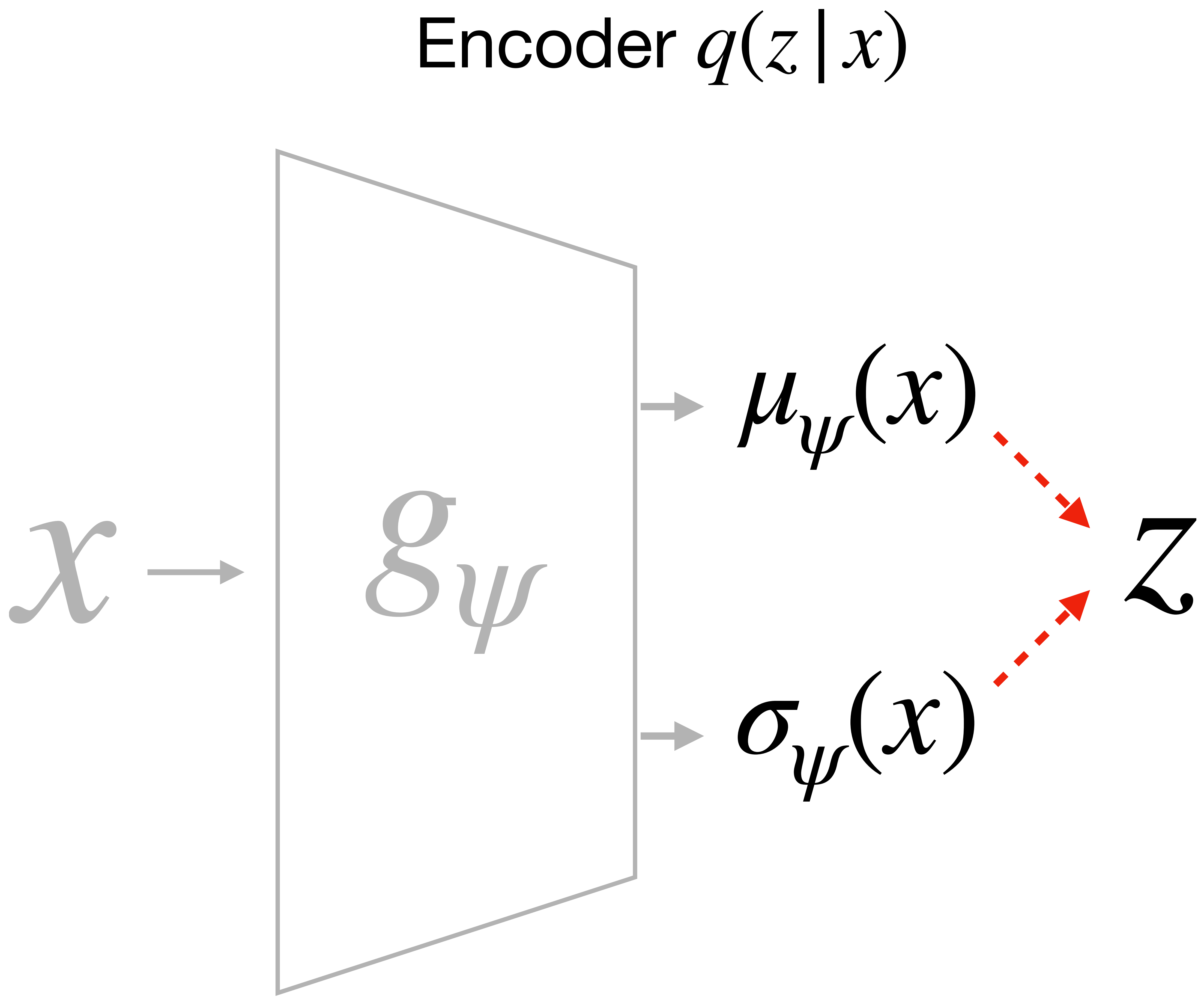
Variational Autoencoders

In our set-up: f_θ is the decoder of a VAE.



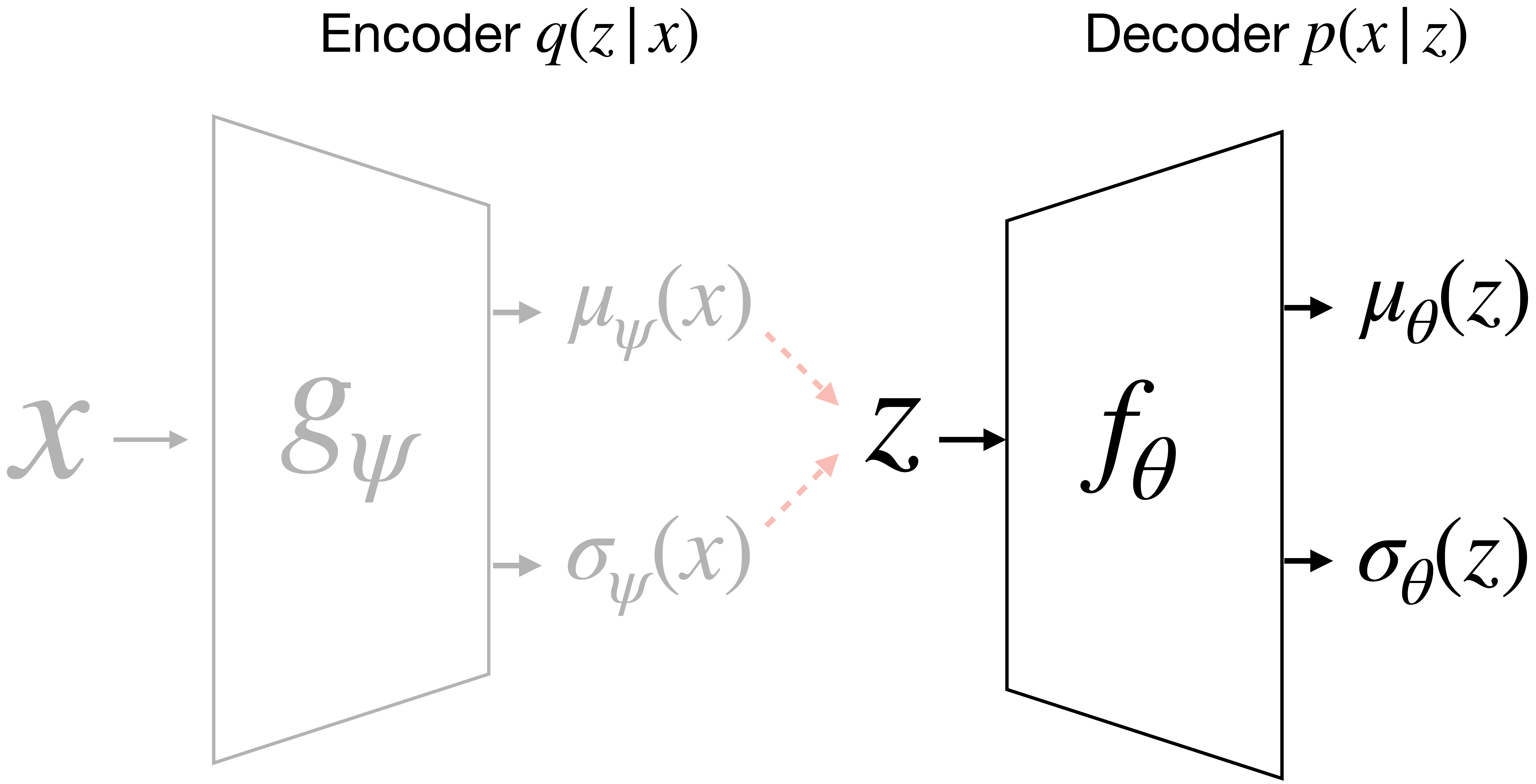
Variational Autoencoders

In our set-up: f_θ is the decoder of a VAE.



Variational Autoencoders

In our set-up: f_θ is the decoder of a VAE.



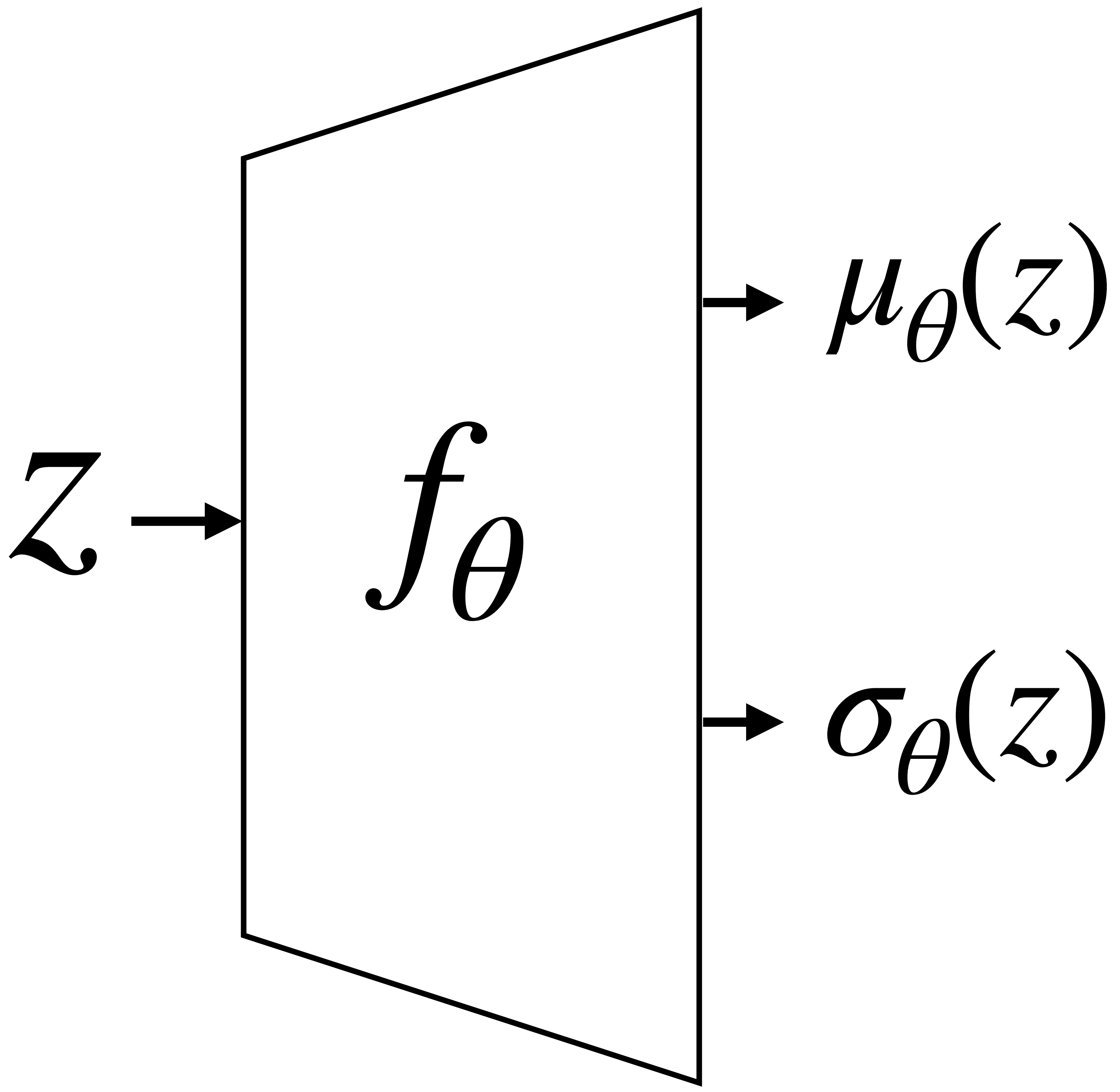
Variational Autoencoders

In our set-up: f_θ is the decoder of a VAE.

Decoder $p(x \mid z)$

Depending on the data, we can decode to a

- Bernoulli (e.g. MNIST)
- Categorical (e.g. strings)
- Normal

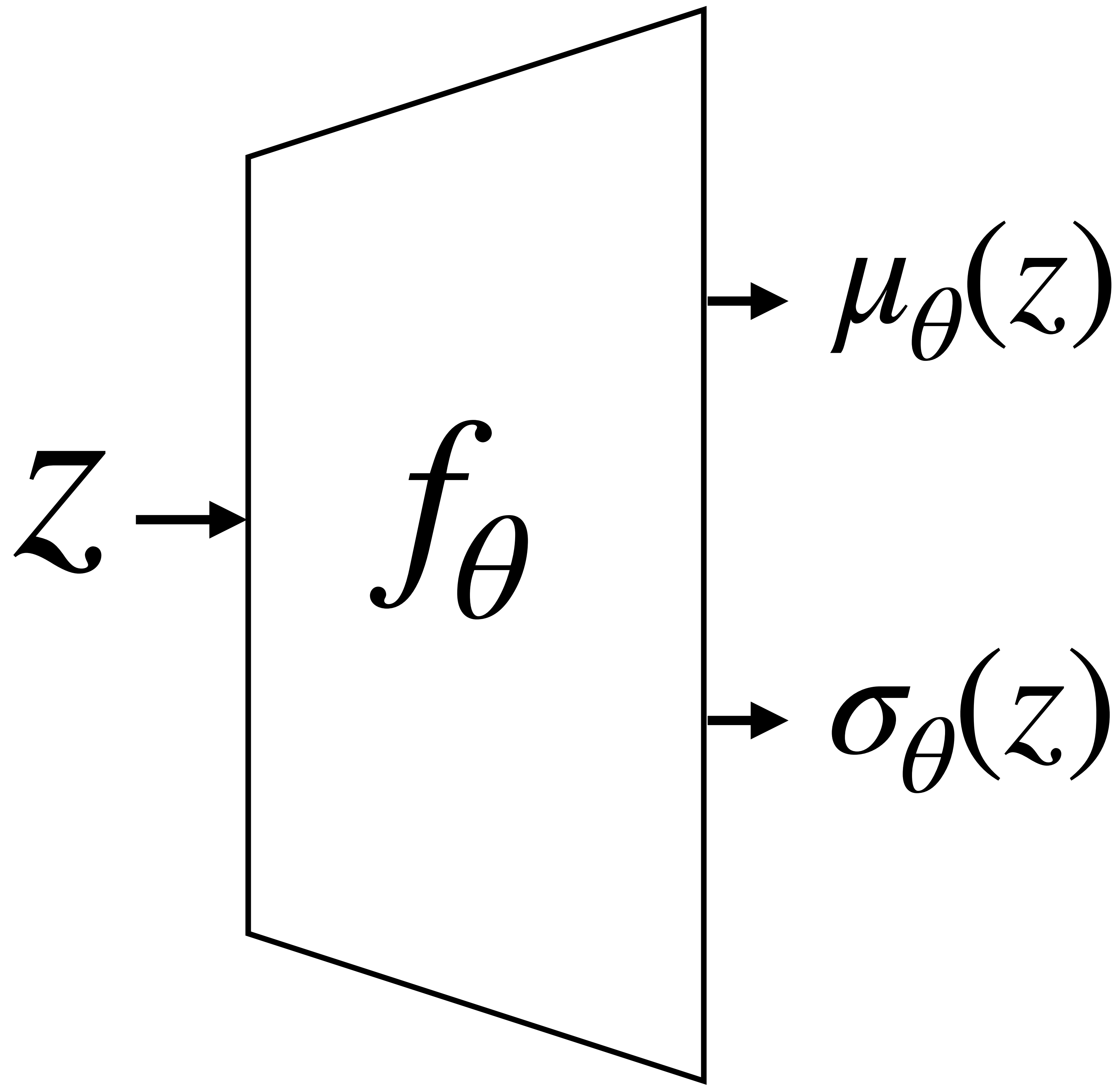


Variational Autoencoders

In our set-up: f_θ is the decoder of a VAE.

LATENT SPACE ODDITY: ON THE CURVATURE OF DEEP GENERATIVE MODELS

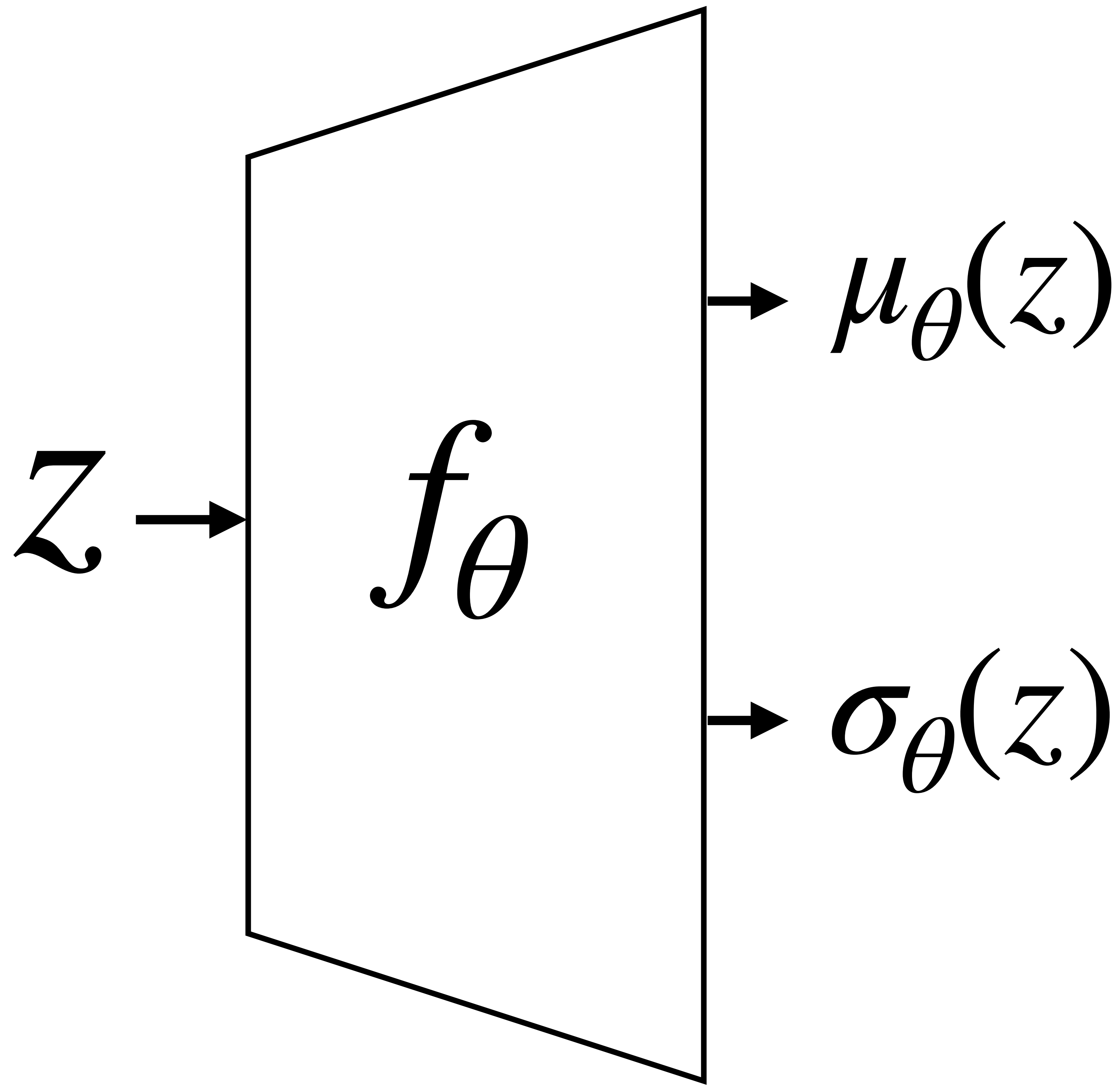
Georgios Arvanitidis, Lars Kai Hansen, Søren Hauberg
Technical University of Denmark, Section for Cognitive Systems
{gear,lkai,sohau}@dtu.dk



Likelihood: Gaussian.

$$f_\theta(z) = \mu_\theta(z) + \epsilon \odot \sigma_\theta(z)^2, \ \ \epsilon \sim N(0, I_D)$$

Our generator is stochastic…



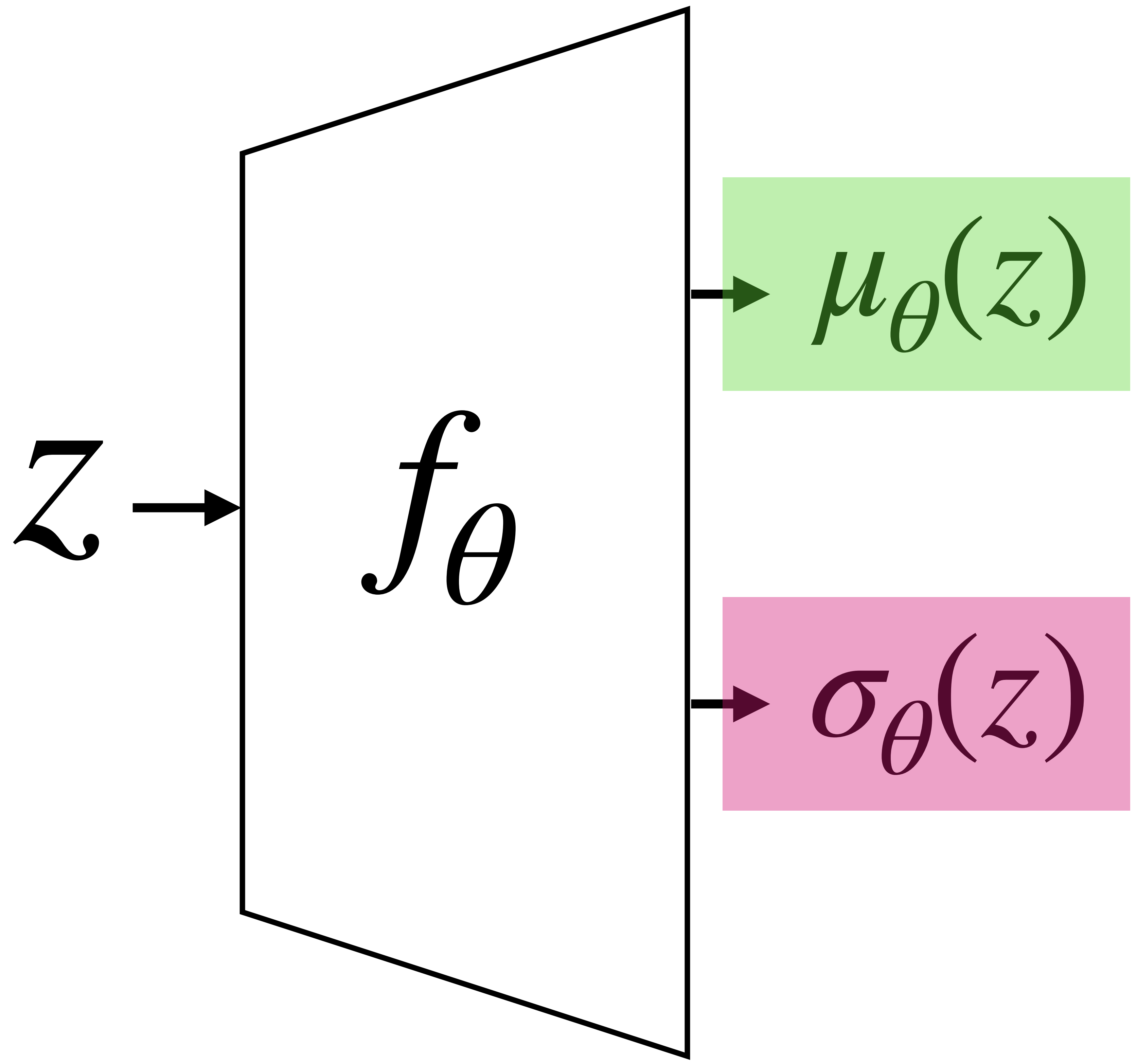
Likelihood: Gaussian.

$$f_\theta(z) = \mu_\theta(z) + \epsilon \odot \sigma_\theta(z)^2, \ \ \epsilon \sim N(0, I_D)$$

Our generator is stochastic...

Theorem 1:

$$\mathbb{E}[J_f(z)^\top J_f(z)] = J_\mu(z)^\top J_\mu(z) + J_\sigma(z)^\top J_\sigma(z)$$



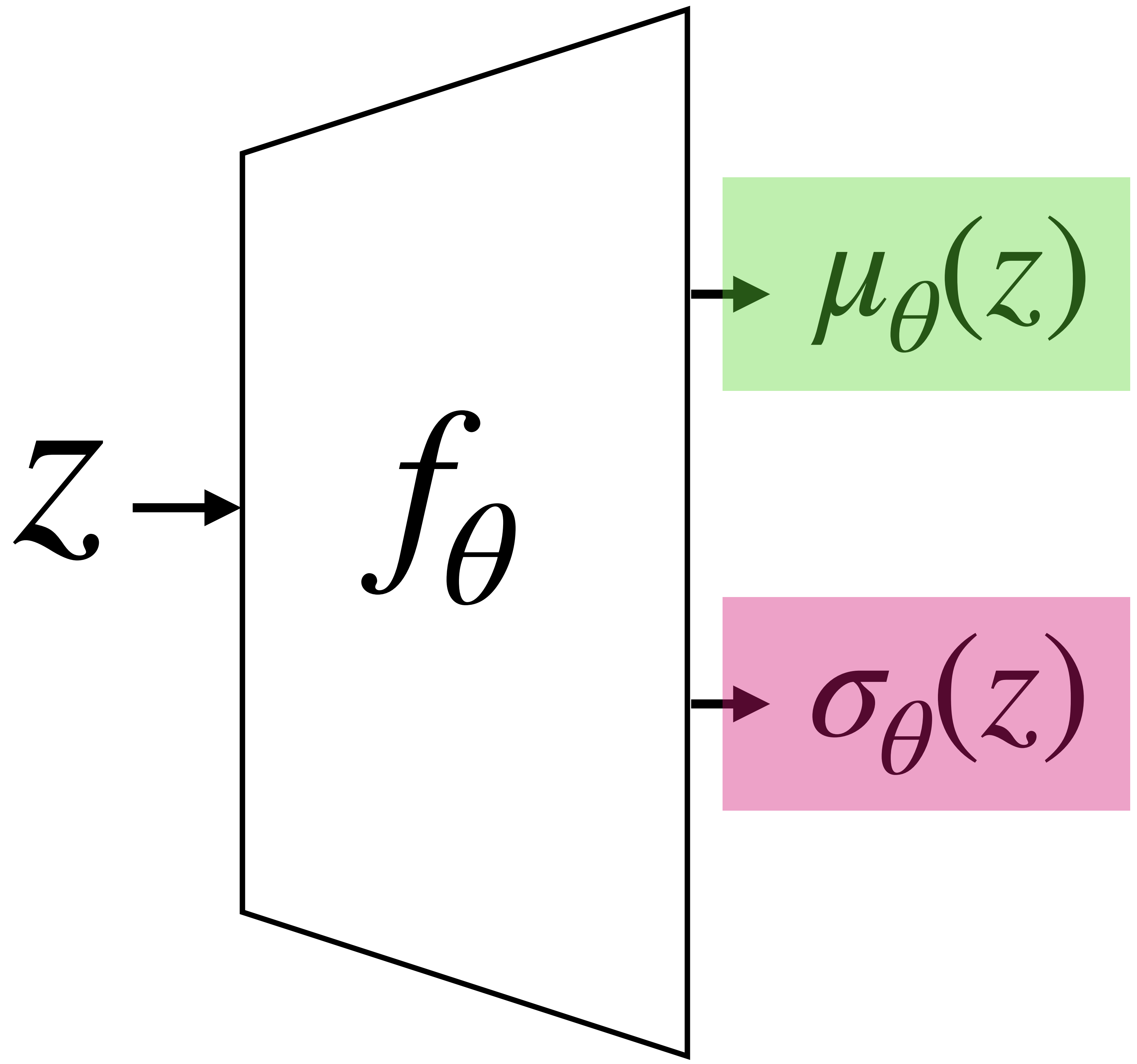
Likelihood: Gaussian.

$$f_\theta(z) = \mu_\theta(z) + \epsilon \odot \sigma_\theta(z)^2, \;\; \epsilon \sim N(0, I_D)$$

Our generator is stochastic…

Theorem 1:

$$\mathbb{E}[J_f(z)^\top J_f(z)] = J_\mu(z)^\top J_\mu(z) + J_\sigma(z)^\top J_\sigma(z)$$



Likelihood: Gaussian.

$$f_\theta(z) = \mu_\theta(z) + \epsilon \odot \sigma_\theta(z)^2, \;\; \epsilon \sim N(0, I_D)$$

Our generator is stochastic…

Theorem 1:

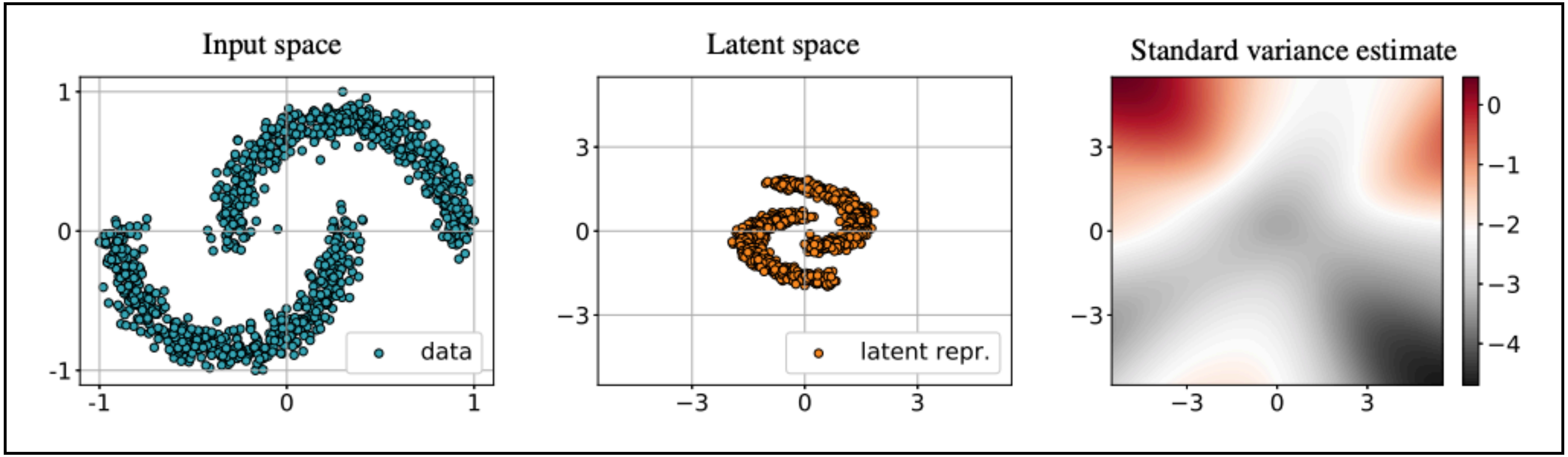
$$\mathbb{E}[J_f(z)^\top J_f(z)] = J_\mu(z)^\top J_\mu(z) + J_\sigma(z)^\top J_\sigma(z)$$

Takeaway: uncertainties are important

LATENT SPACE ODDITY

Takeaway: uncertainties are important

$$\sigma_\theta(z)$$



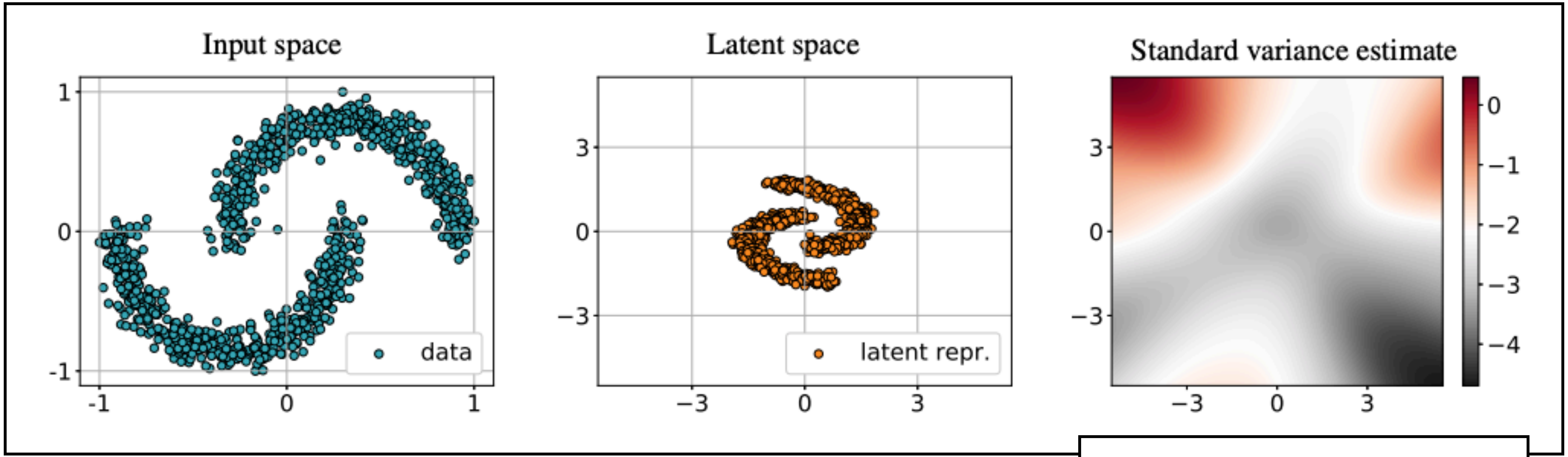
Unfortunately…

VAEs don't have good uncertainty quantification out-of-the-box

LATENT SPACE ODDITY

Takeaway: uncertainties are important

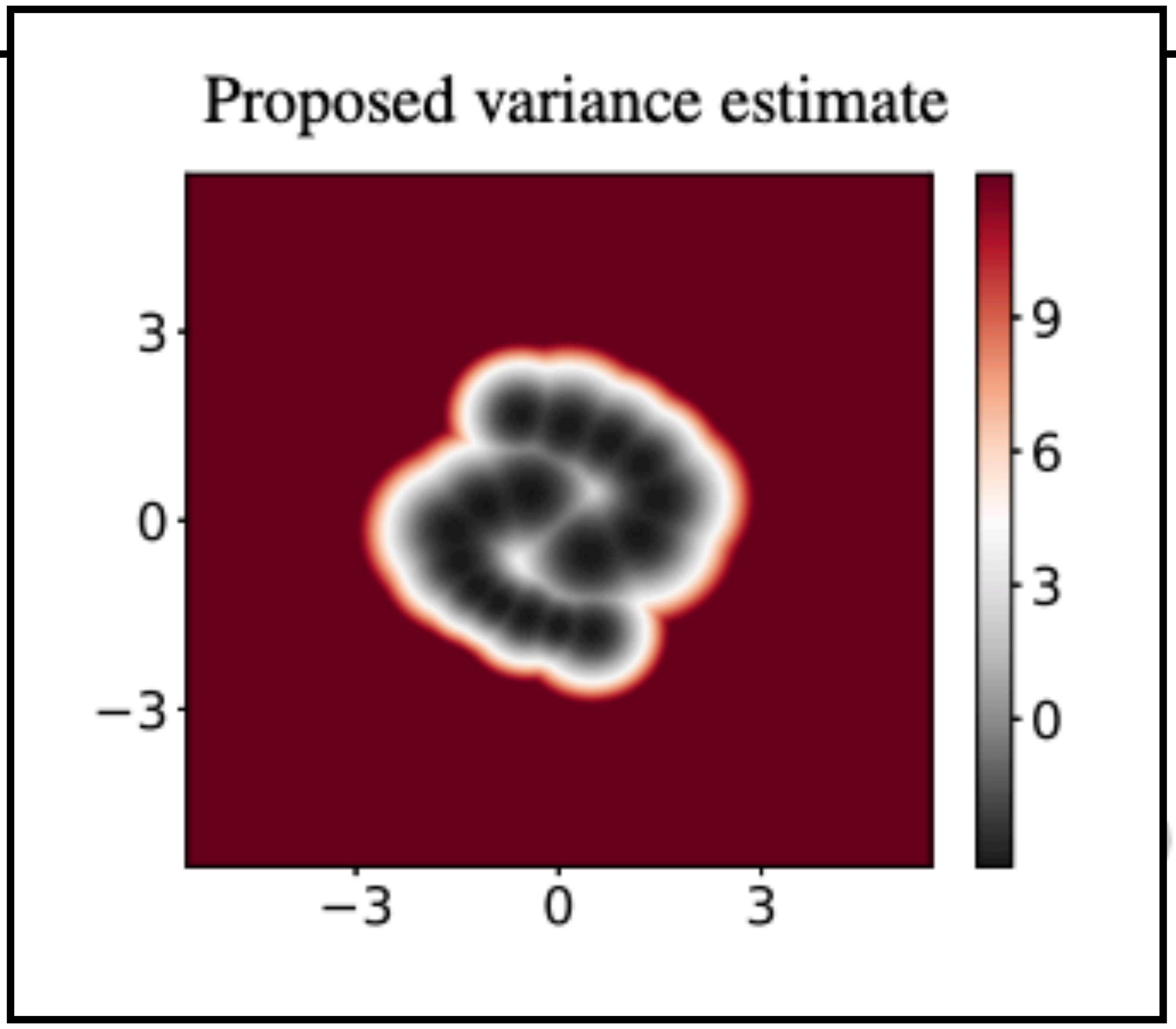
$\sigma_\theta(z)$



Unfortunately…

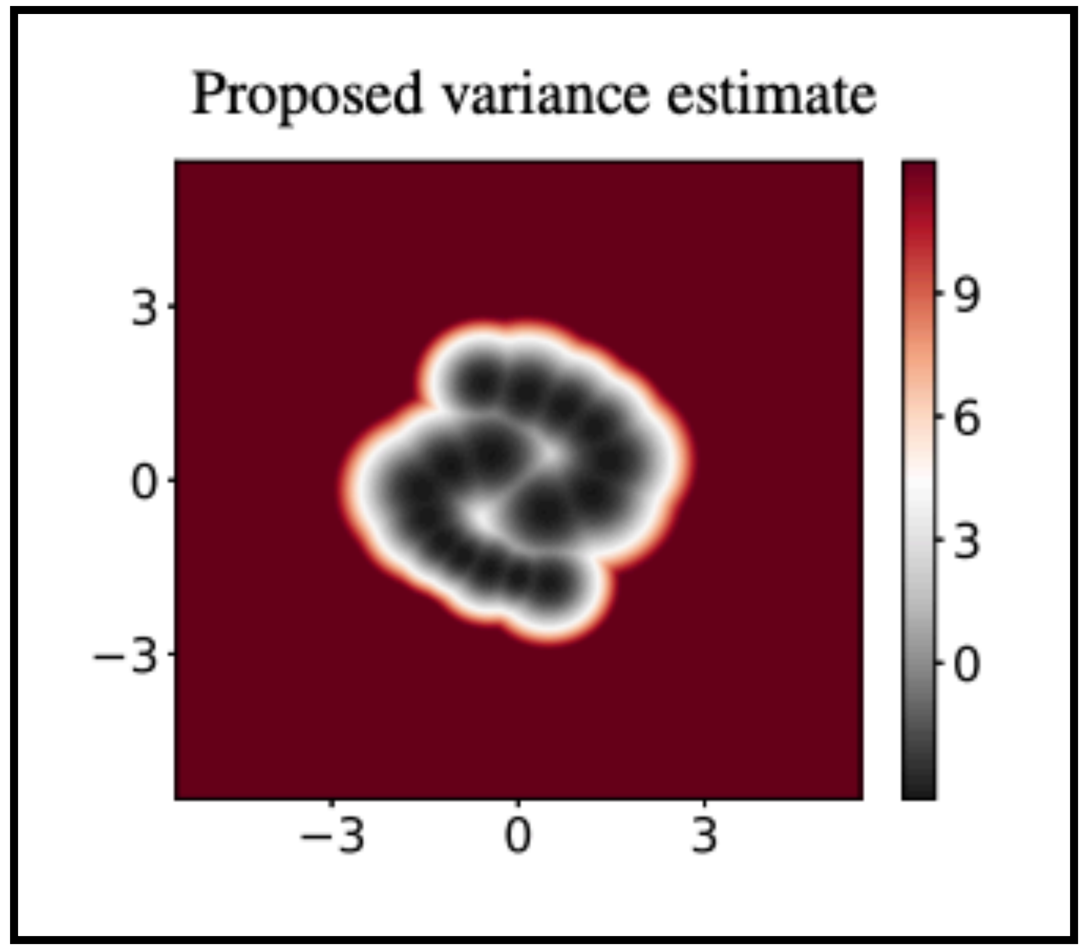
VAEs don't have good uncertainty quantification out-of-the-box

Ideally…



Takeaway: uncertainties are important

How to *calibrate* the uncertainty?



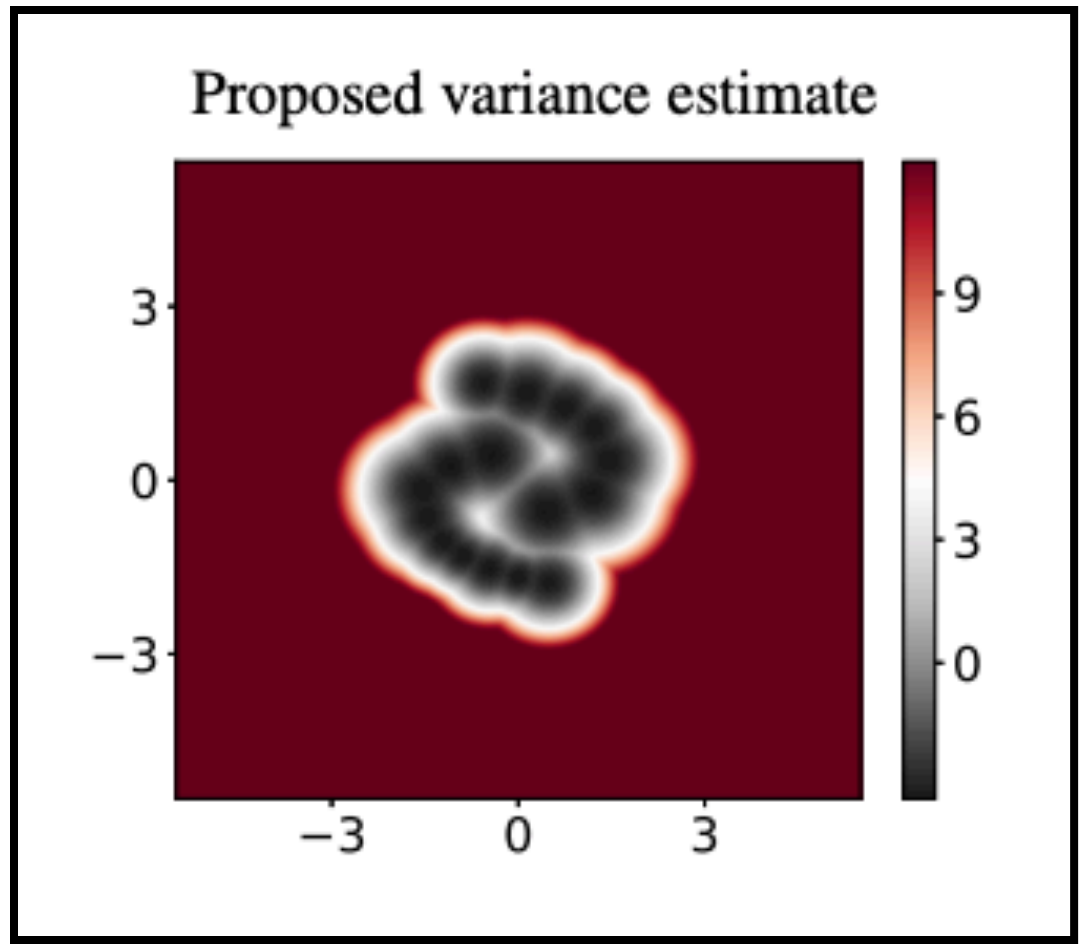
Overwrite $\sigma_\theta(z)$ like this

$$\tilde{\sigma}_\theta(z) = \begin{cases} \sigma_\theta(z) & \text{if } z \text{ is close to the training codes,} \\ \text{a large number} & \text{otherwise.} \end{cases}$$

LATENT SPACE ODDITY

Takeaway: uncertainties are important

Contrastive Divergence



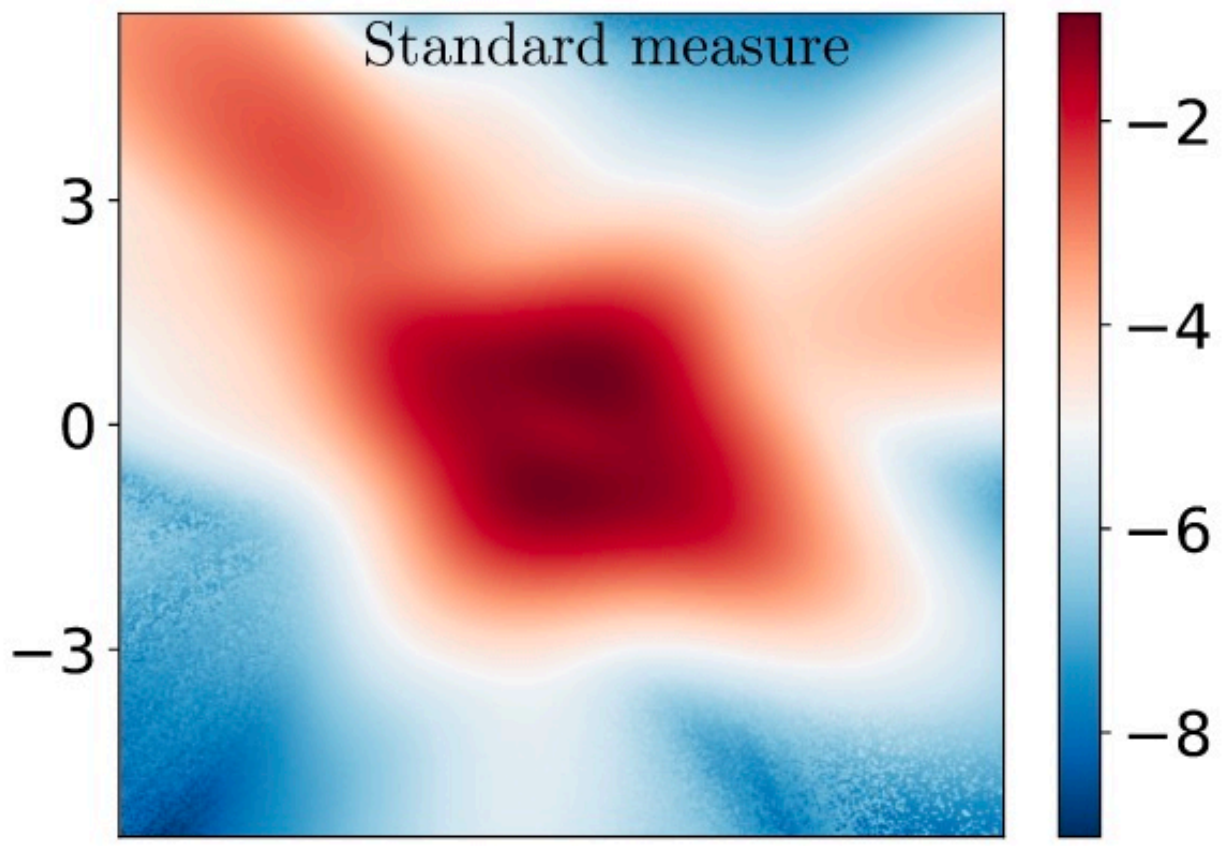
How to *calibrate* the uncertainty?

Overwrite $\sigma_\theta(z)$ like this

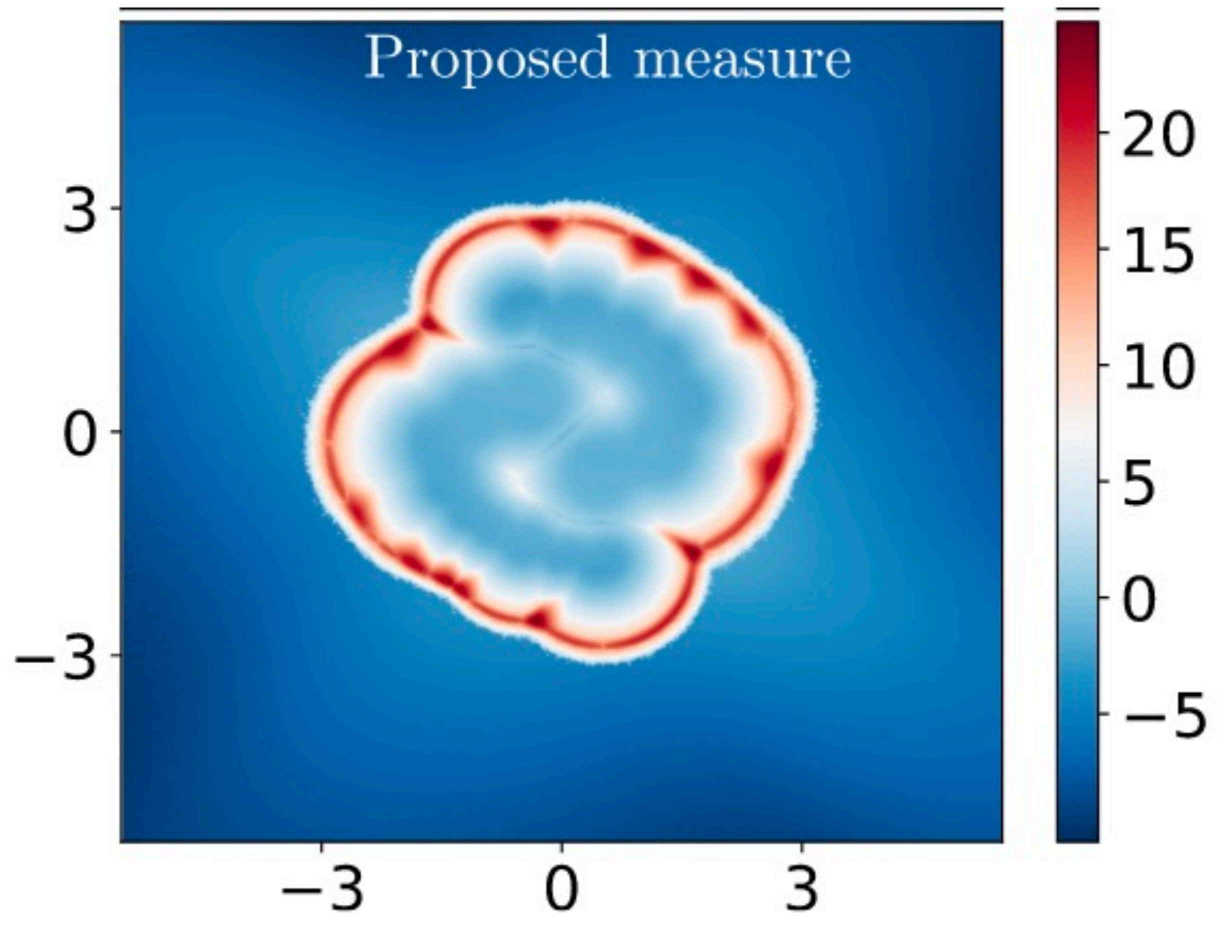
$$\tilde{\sigma}_\theta(z) = \begin{cases} \sigma_\theta(z) & \text{if } z \text{ is close to the training codes,} \\ \text{a large number} & \text{otherwise.} \end{cases}$$

$$\mathbb{E}[J_f(z)^\top J_f(z)]$$

Volume before



Volume after

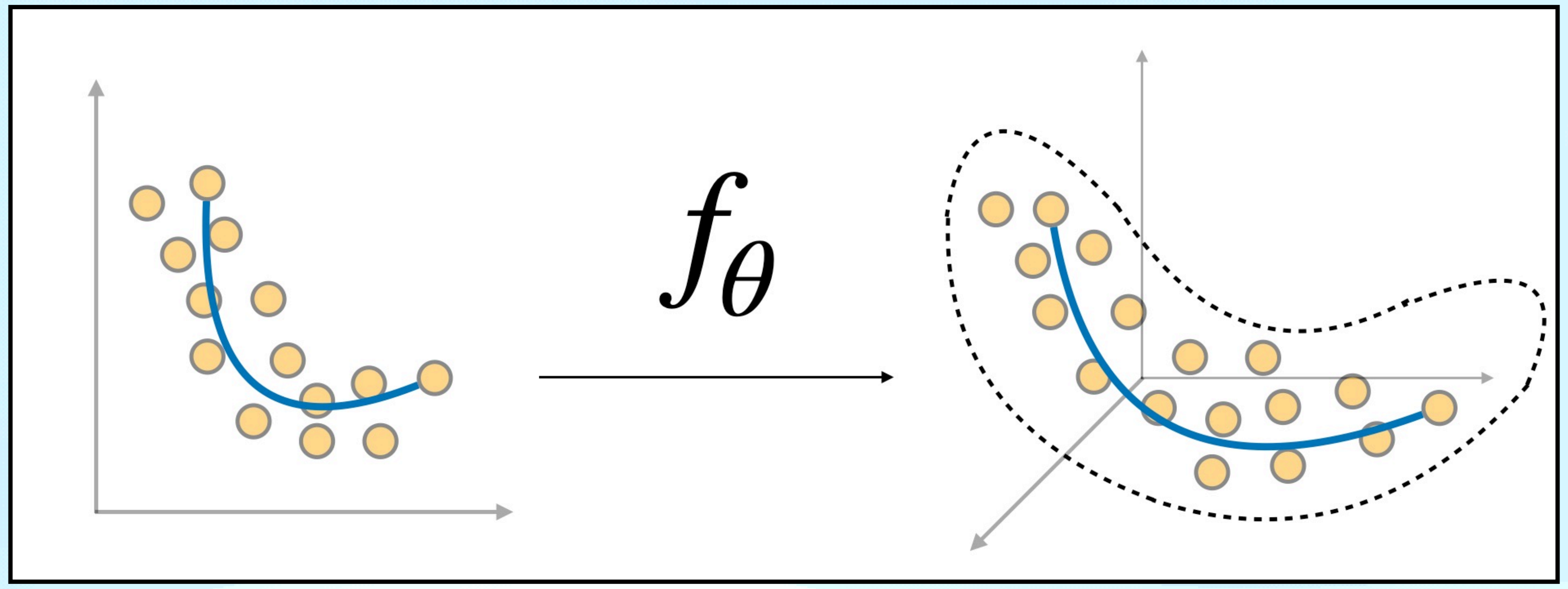


$$\text{volume}(z) = \sqrt{\det(J^\top J)}$$

LATENT SPACE ODDITY

Summary of the set-up

We learn latent representations using VAEs, and use the decoder to pull back geometry



Summary of the set-up

We learn latent representations using VAEs, and use the decoder to pull back geometry

The pullback metric is highly dependent on the likelihood we choose

$$J_\mu(z)^\top J_\mu(z) + J_\sigma(z)^\top J_\sigma(z)$$

Summary of the set-up

We learn latent representations using VAEs, and use the decoder to pull back geometry

The pullback metric is highly dependent on the likelihood we choose

$$J_\mu(z)^\top J_\mu(z) + J_\sigma(z)^\top J_\sigma(z)$$

Uncertainty quantification is important, and we currently do it by hand.

$$\tilde{\sigma}_\theta(z) = \begin{cases} \sigma_\theta(z) \\ \text{a large number} \end{cases}$$

Some applications

Learning Riemannian Manifolds for Geodesic Motion Skills

Hadi Beik-Mohammadi[1,2], Søren Hauberg[3], Georgios Arvanitidis[4], Gerhard Neumann[2], and Leonel Rozo[1]



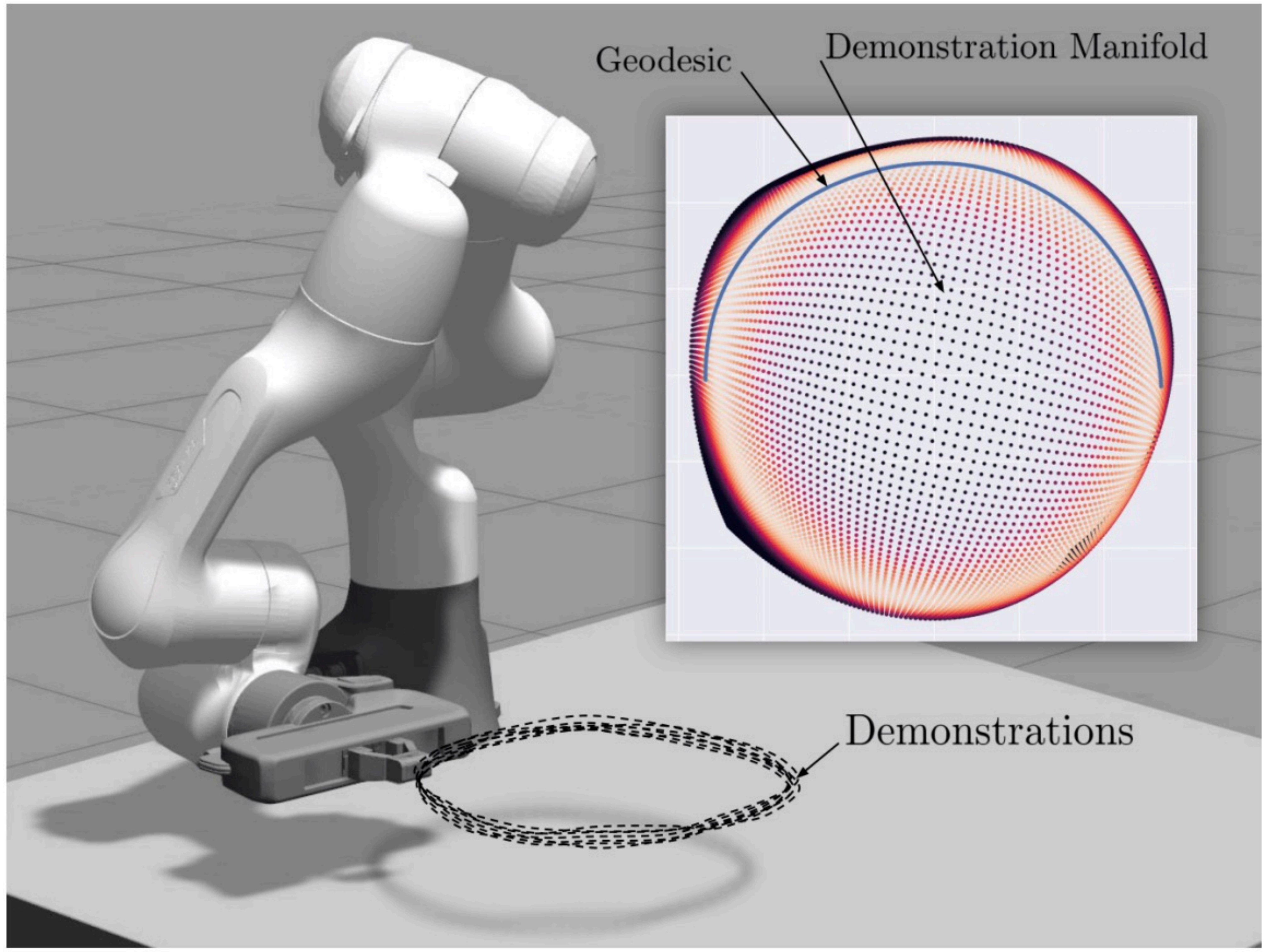



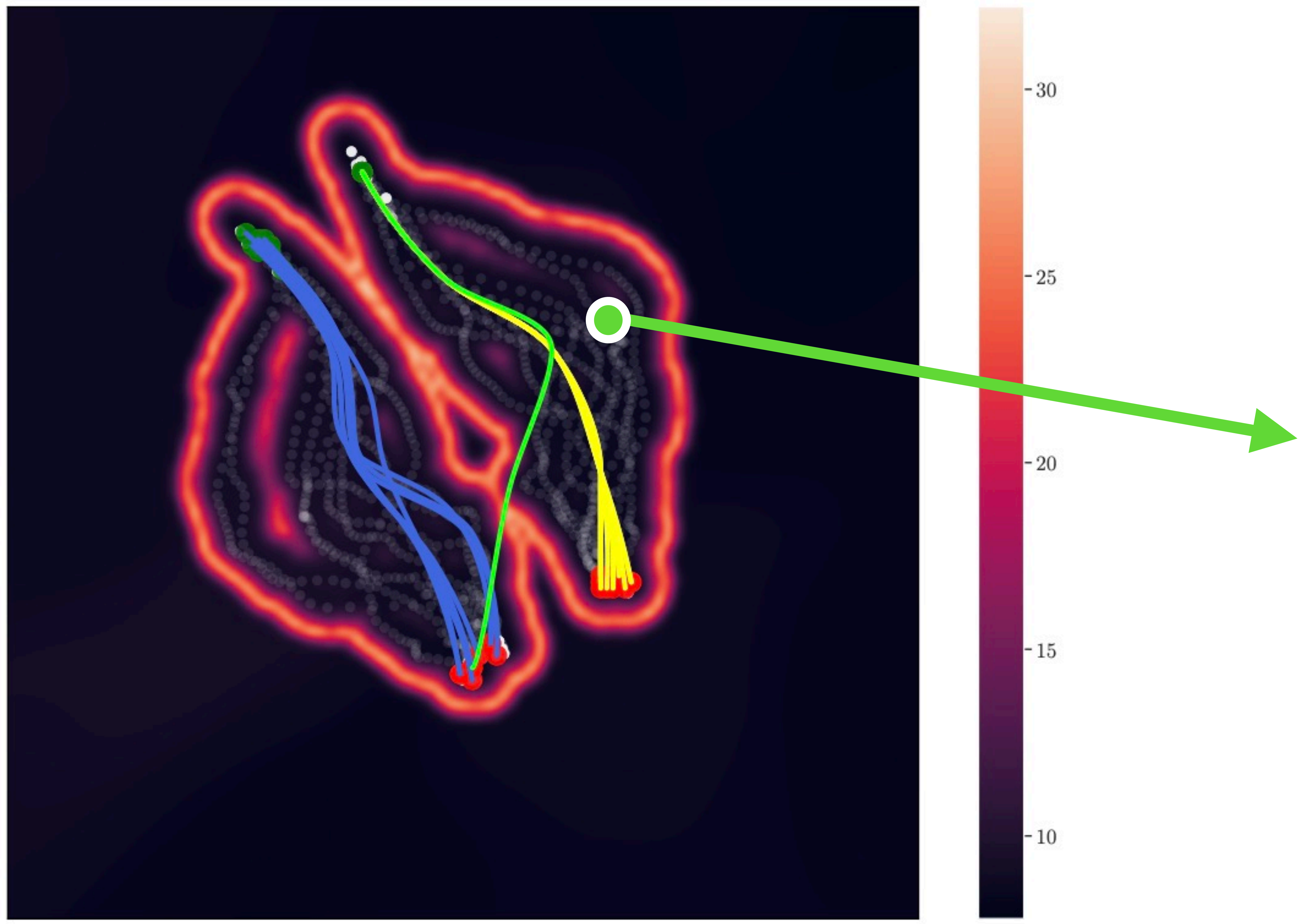
Fig. 1: From demonstrations we learn a variational autoencoder that spans a random Riemannian manifold. Geodesics on this manifold are viewed as motion skills.

Data: Demonstrations of a robot task

$$(p, r) \in \mathbb{R}^3 \times \mathbb{S}^3$$

Goal: Learn a joint latent space, and control through geodesics

Learning Riemannian Manifolds for Geodesic Motion Skills

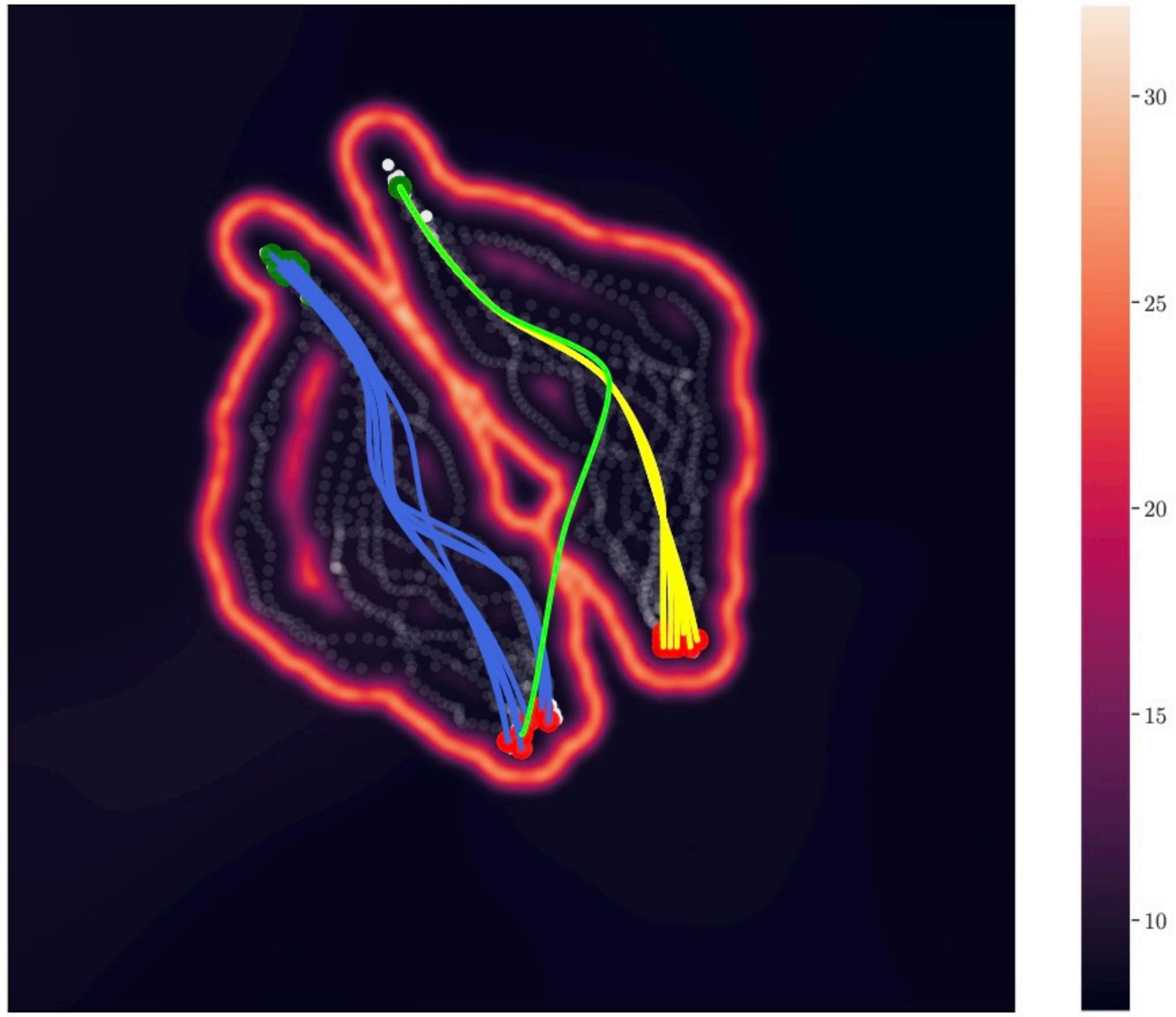


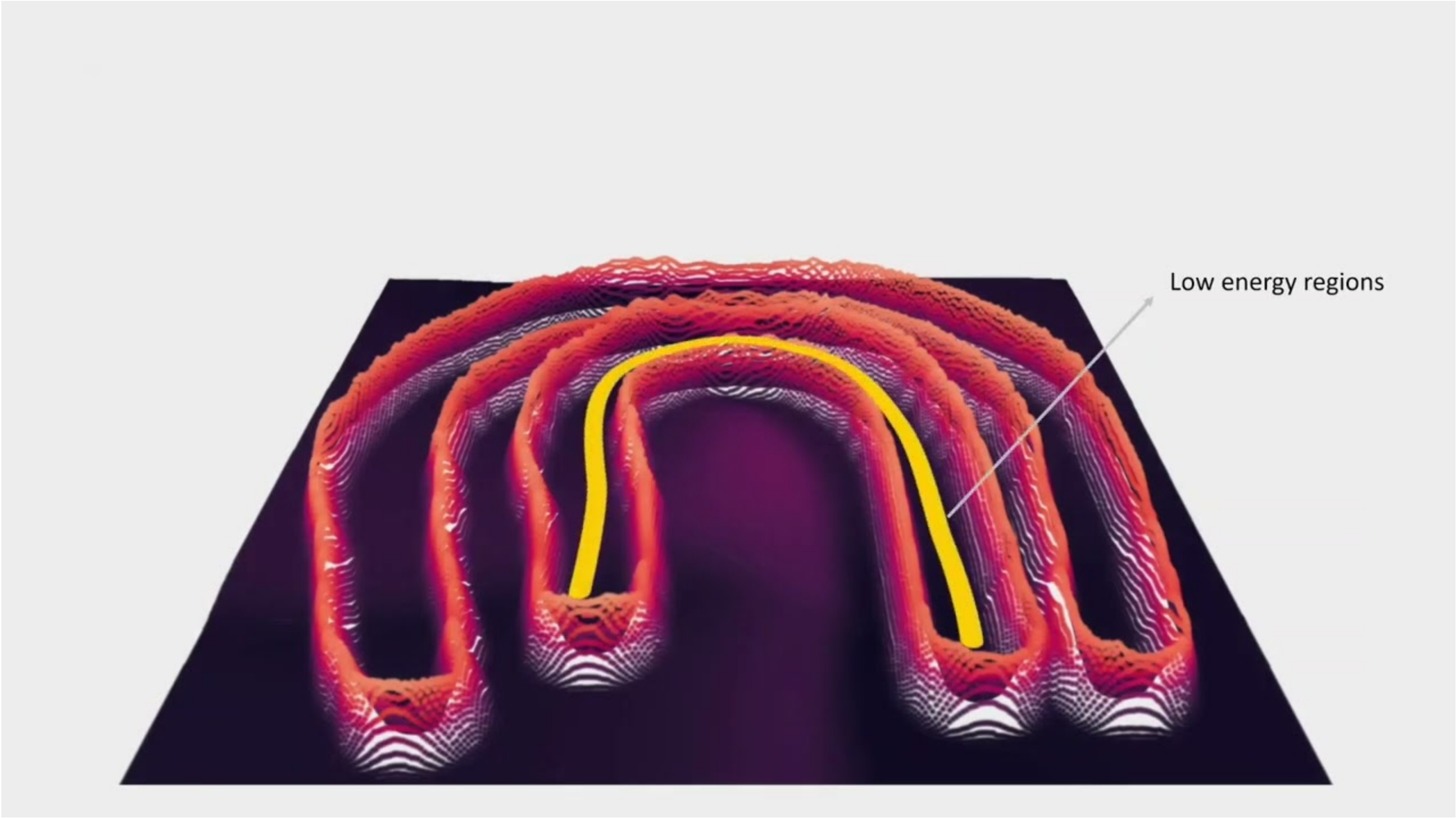
Each point in latent space corresponds to a robot arm configuration

Goal: Learn a joint latent space, and control through geodesics

Learning Riemannian Manifolds for Geodesic Motion Skills

Timestamp: 2:23



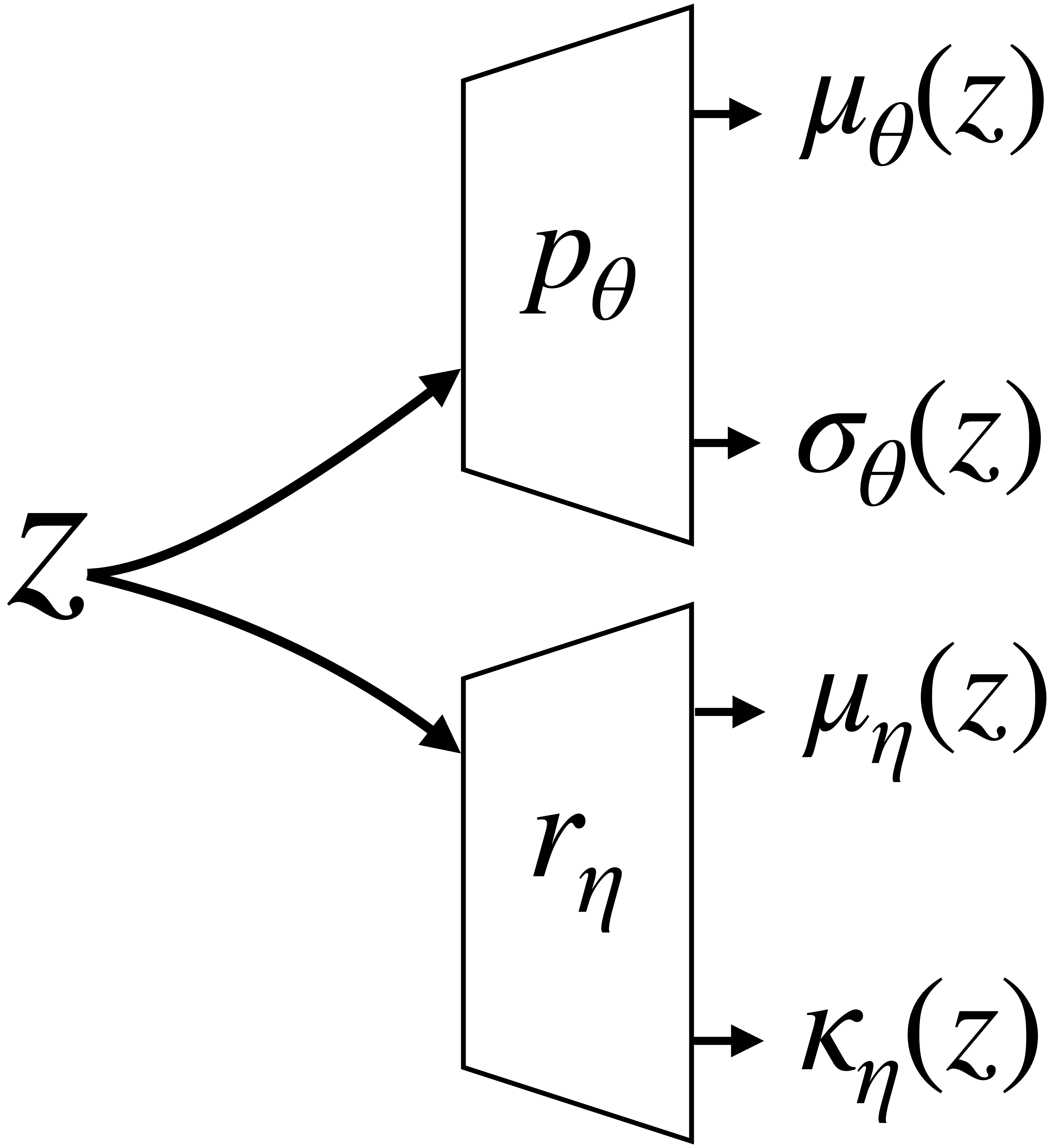


Goal: Learn a joint latent space, and control through geodesics

Learning Riemannian Manifolds for Geodesic Motion Skills

Data: Demonstrations of a robot task

$(p, r) \in \mathbb{R}^3 \times \mathbb{S}^3$



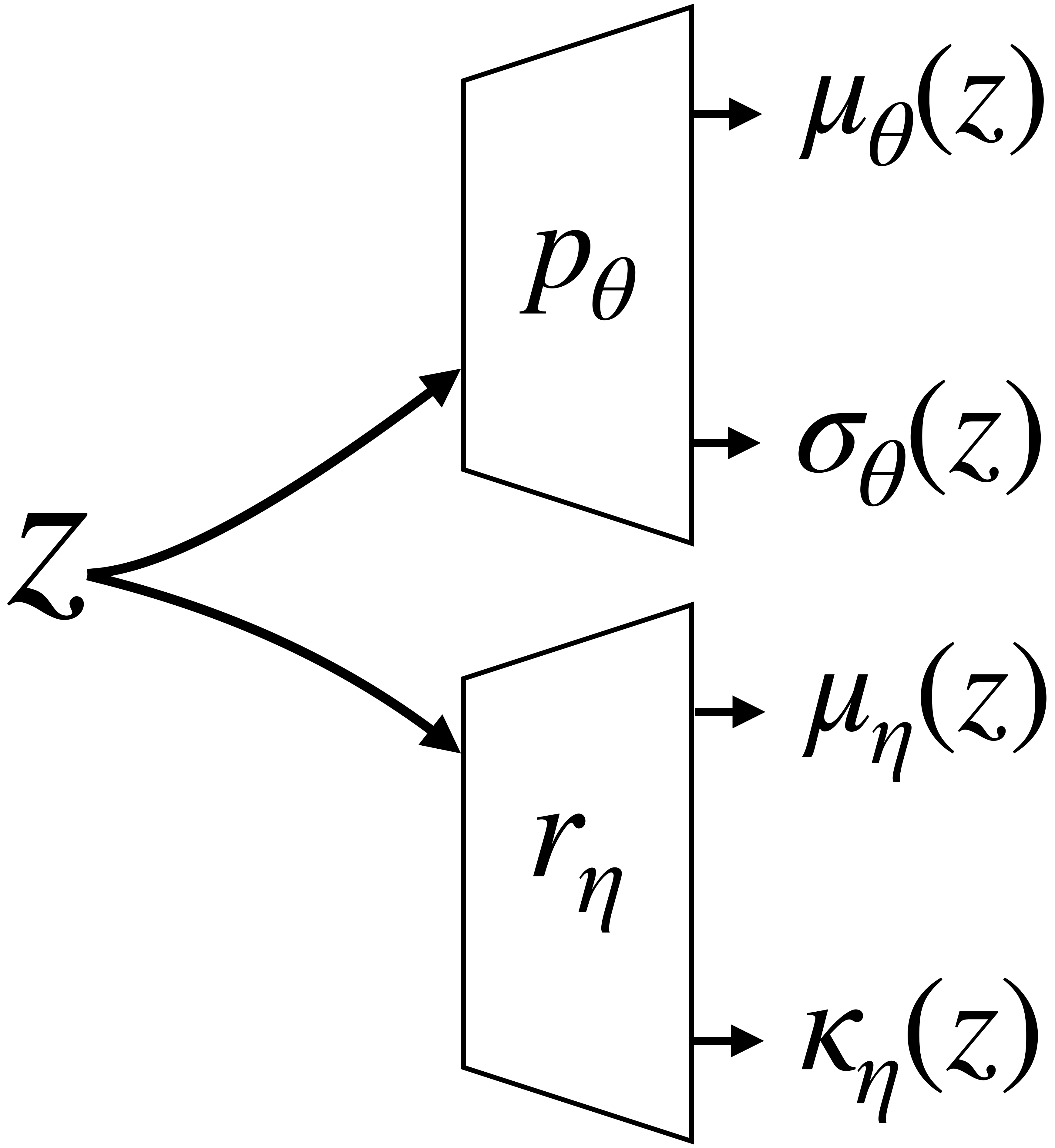
They learn a Gaussian for positions, and a von Mises-Fisher for rotations…

$$r \sim \text{vMF}(r \mid \mu, \kappa) \Rightarrow r \in \mathbb{S}^3$$

Learning Riemannian Manifolds for Geodesic Motion Skills

Data: Demonstrations of a robot task

$$(p, r) \in \mathbb{R}^3 \times \mathbb{S}^3$$



They learn a Gaussian for positions, and a von Mises-Fisher for rotations…

$$r \sim \text{vMF}(r \,|\, \mu, \kappa) \Rightarrow r \in \mathbb{S}^3$$

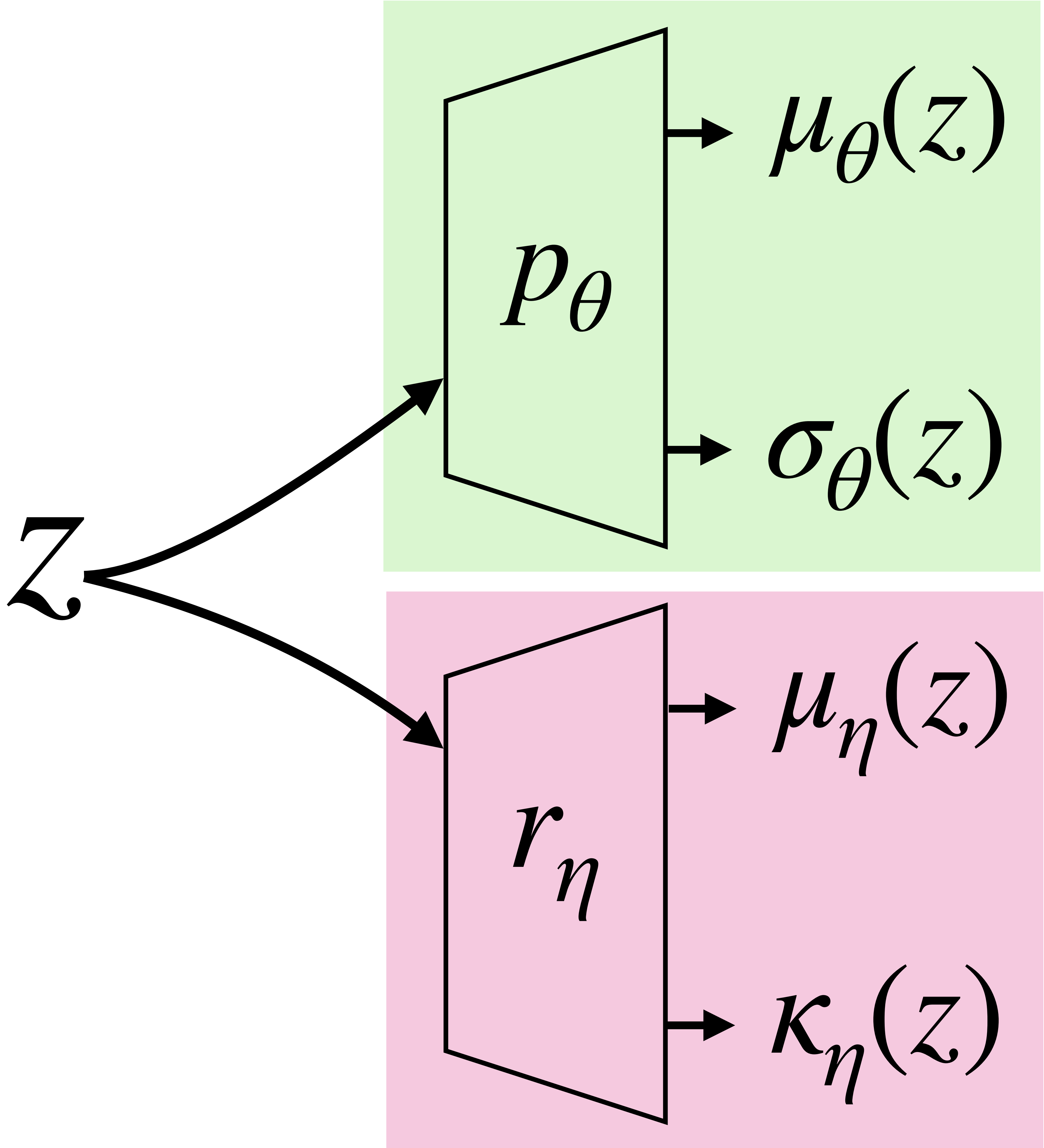
The expected pullback metric also has closed form…

$$J_\mu^p(z)^\top J_\mu^p(z) + J_\sigma^p(z)^\top J_\sigma^p(z)$$
$$+ J_\mu^r(z)^\top J_\mu^r(z) + J_\kappa^r(z)^\top J_\kappa^r(z)$$

Learning Riemannian Manifolds for Geodesic Motion Skills

Data: Demonstrations of a robot task

$(p, r) \in \mathbb{R}^3 \times \mathbb{S}^3$



They learn a Gaussian for positions, and a von Mises-Fisher for rotations…

$$r \sim \text{vMF}(r \,|\, \mu, \kappa) \Rightarrow r \in \mathbb{S}^3$$

The expected pullback metric also has closed form…

$$J^p_\mu(z)^\top J^p_\mu(z) + J^p_\sigma(z)^\top J^p_\sigma(z)$$

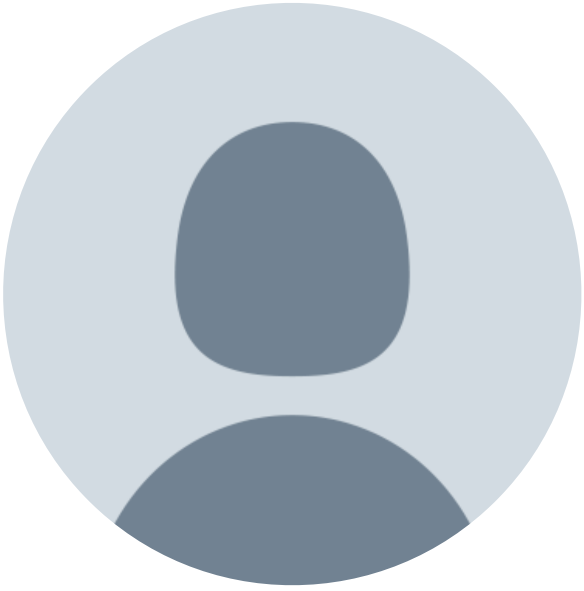
$$+ J^r_\mu(z)^\top J^r_\mu(z) + J^r_\kappa(z)^\top J^r_\kappa(z)$$

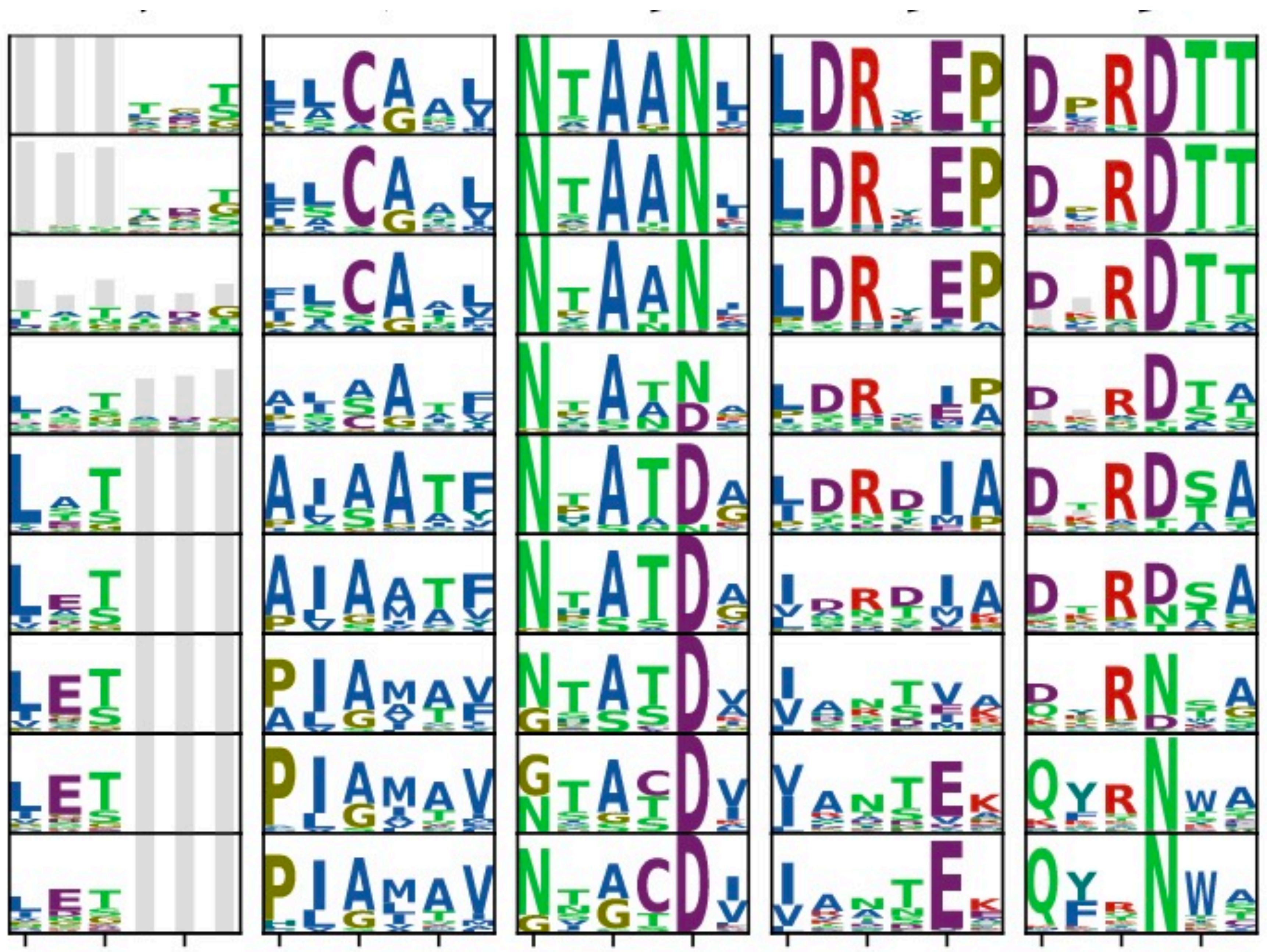
Learning Riemannian Manifolds for Geodesic Motion Skills

Article | Open Access | Published: 08 April 2022

Learning meaningful representations of protein sequences

Nicki Skafte Detlefsen, Søren Hauberg & Wouter Boomsma



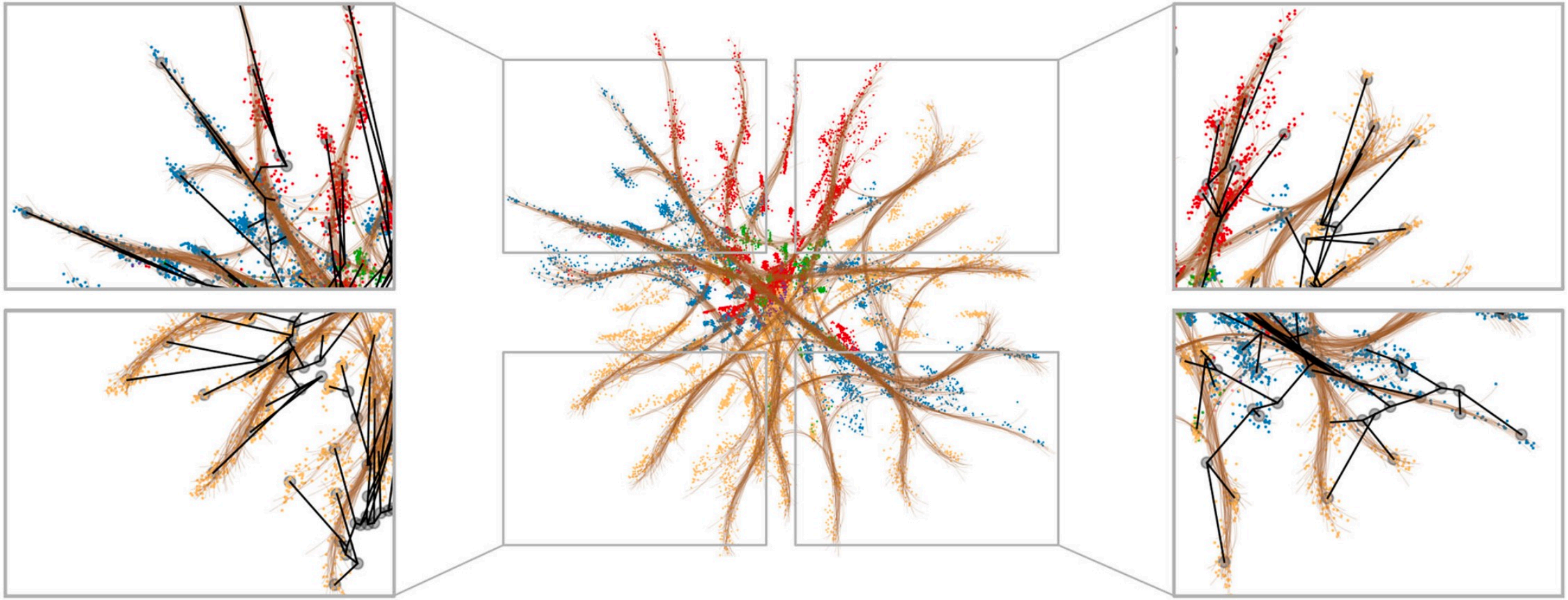


Data: Protein sequences (i.e. strings)

Goal: Build meaningful representations

Learning meaningful representations of protein sequences

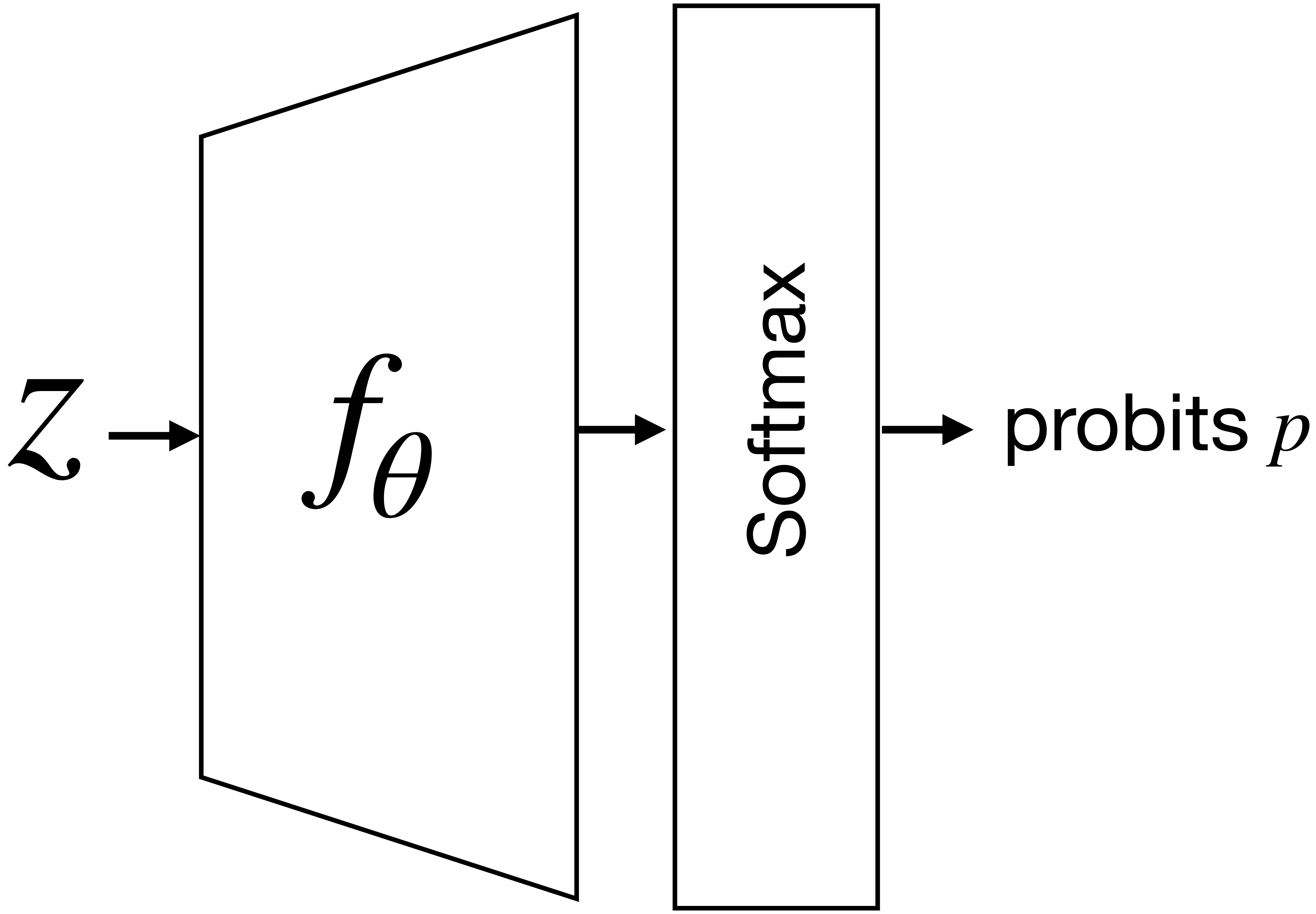
Goal: Build meaningful representations



Geodesics follow the evolution of a protein family!*

Learning meaningful representations of protein sequences

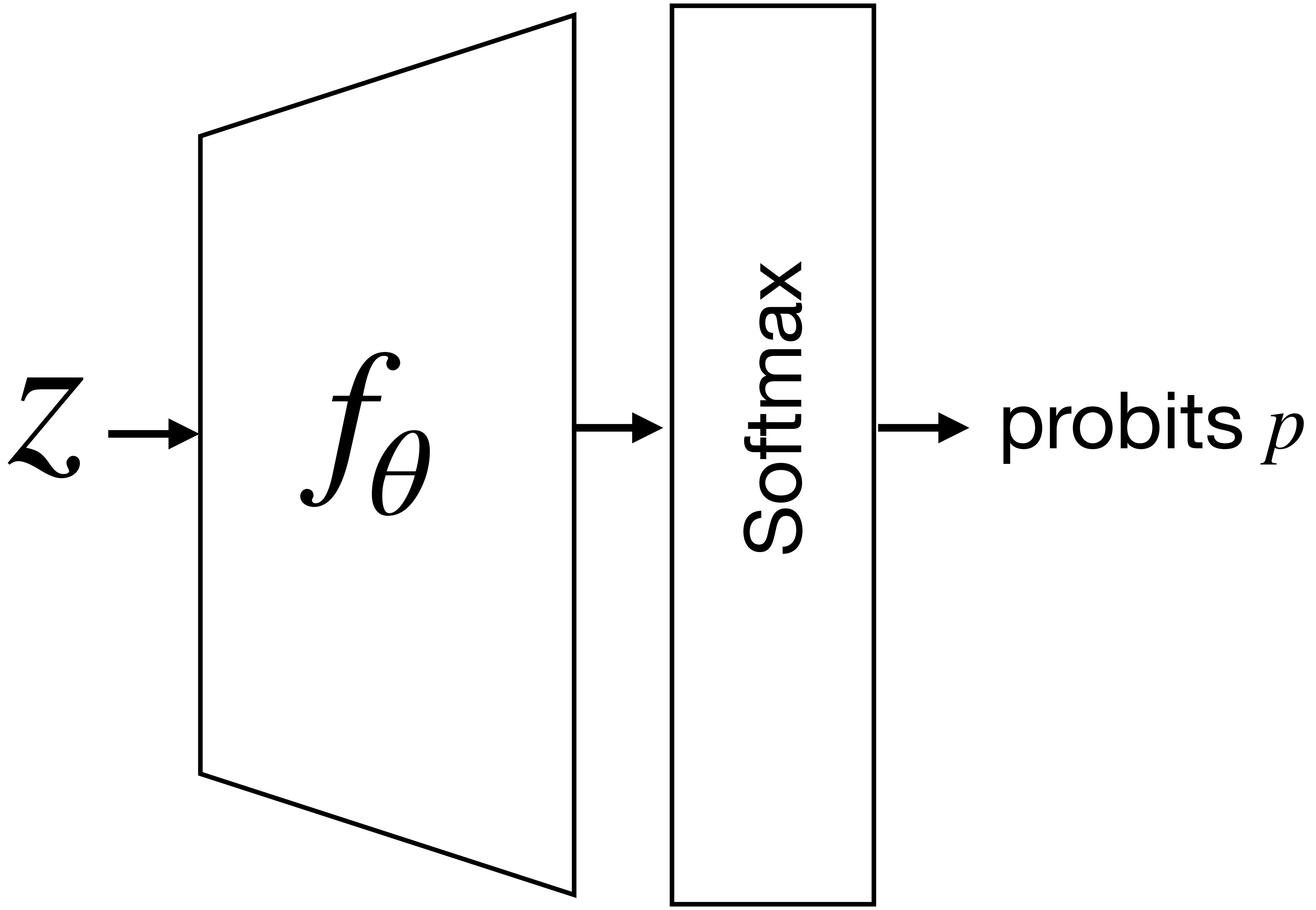
Data: Protein sequences (i.e. strings)



They train a VAE as you would for strings…

Learning meaningful representations of protein sequences

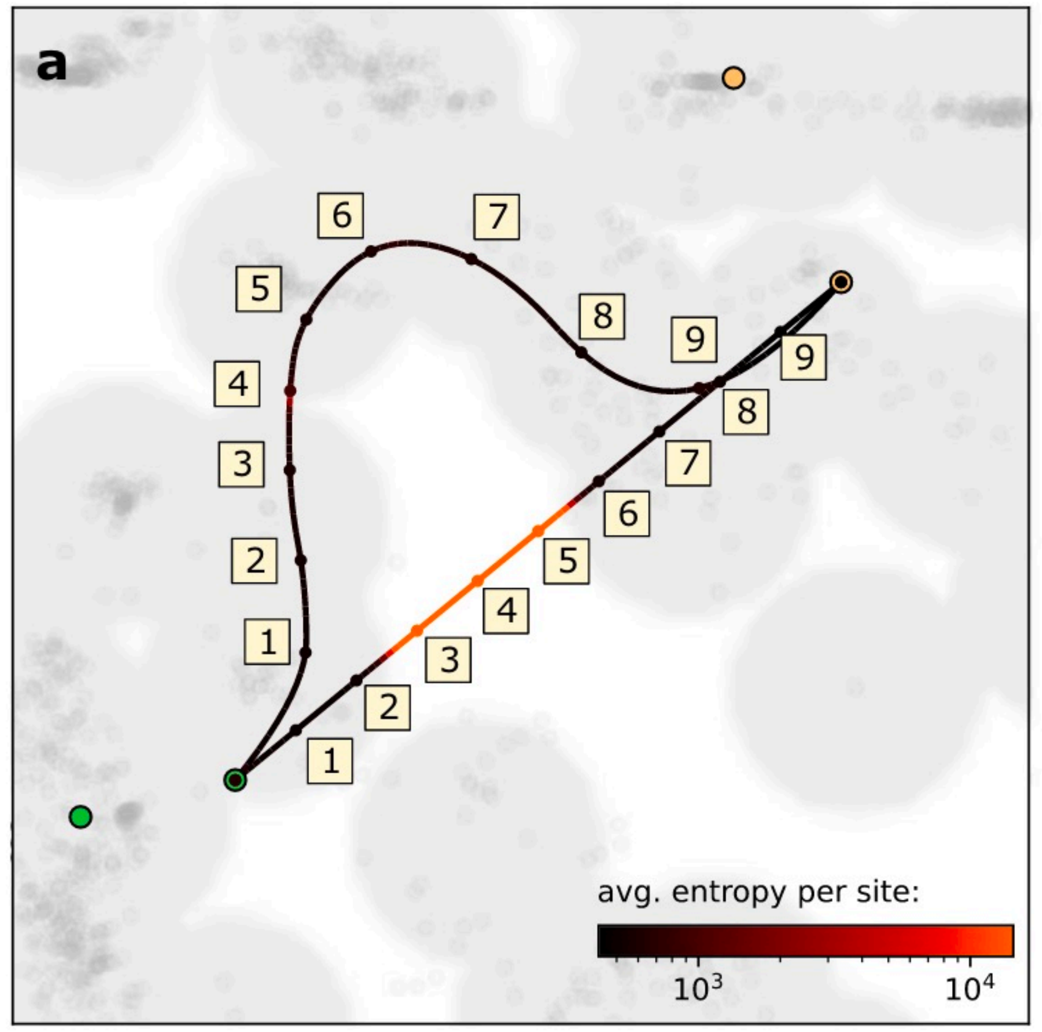
Data: Protein sequences (i.e. strings)



They train a VAE as you would for strings…

…instead of pulling back the metric, they minimize energy of curves.

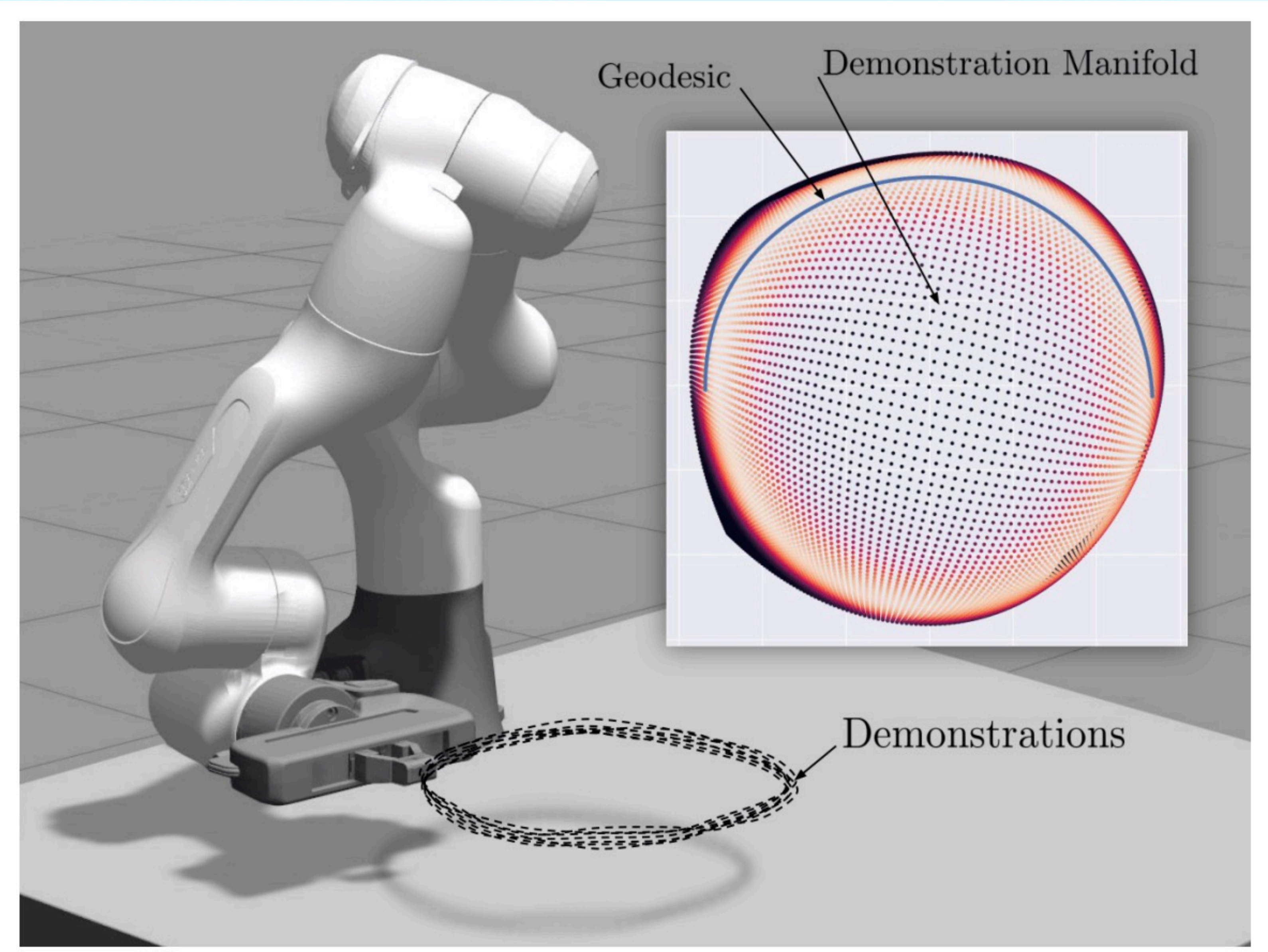
$$\text{Energy}[c] = \sum_t \|p_{t+1} - p_t\|^2$$



Learni … ntations of protein seque

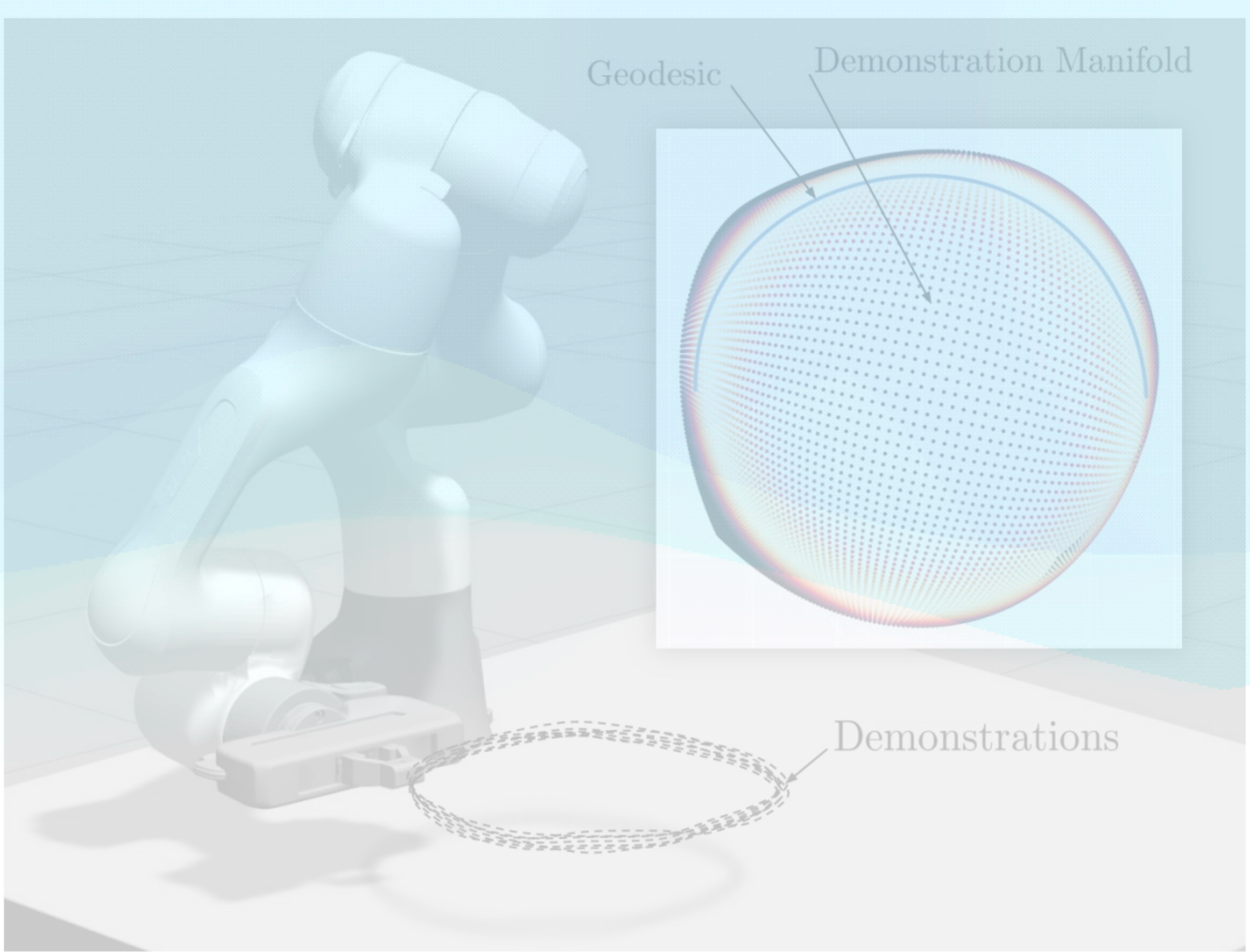
Summary of applications

Latent space geometries have been applied to motion synthesis and protein modeling



Summary of applications

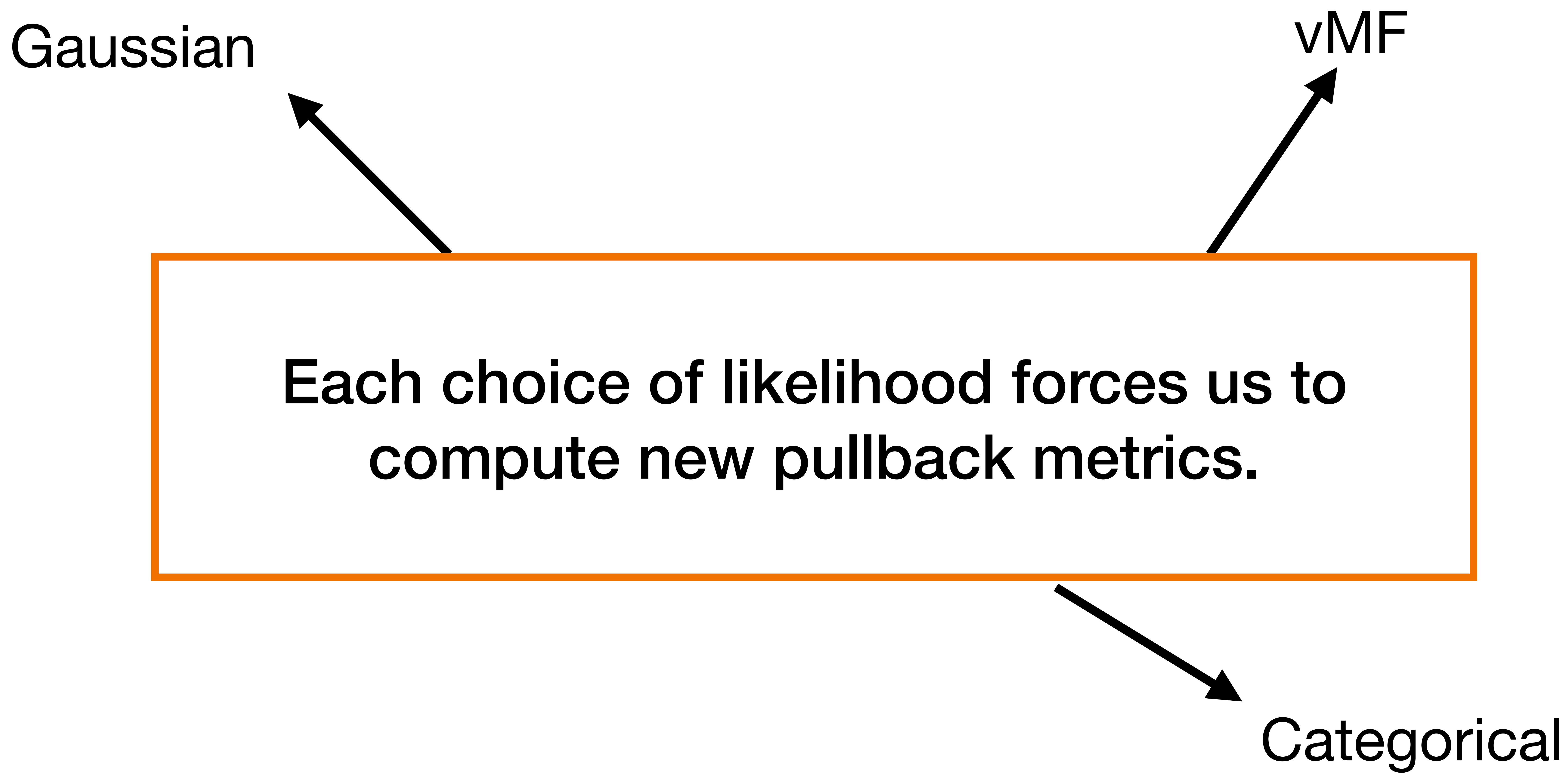
Latent space geometries have been applied to motion synthesis and protein modeling



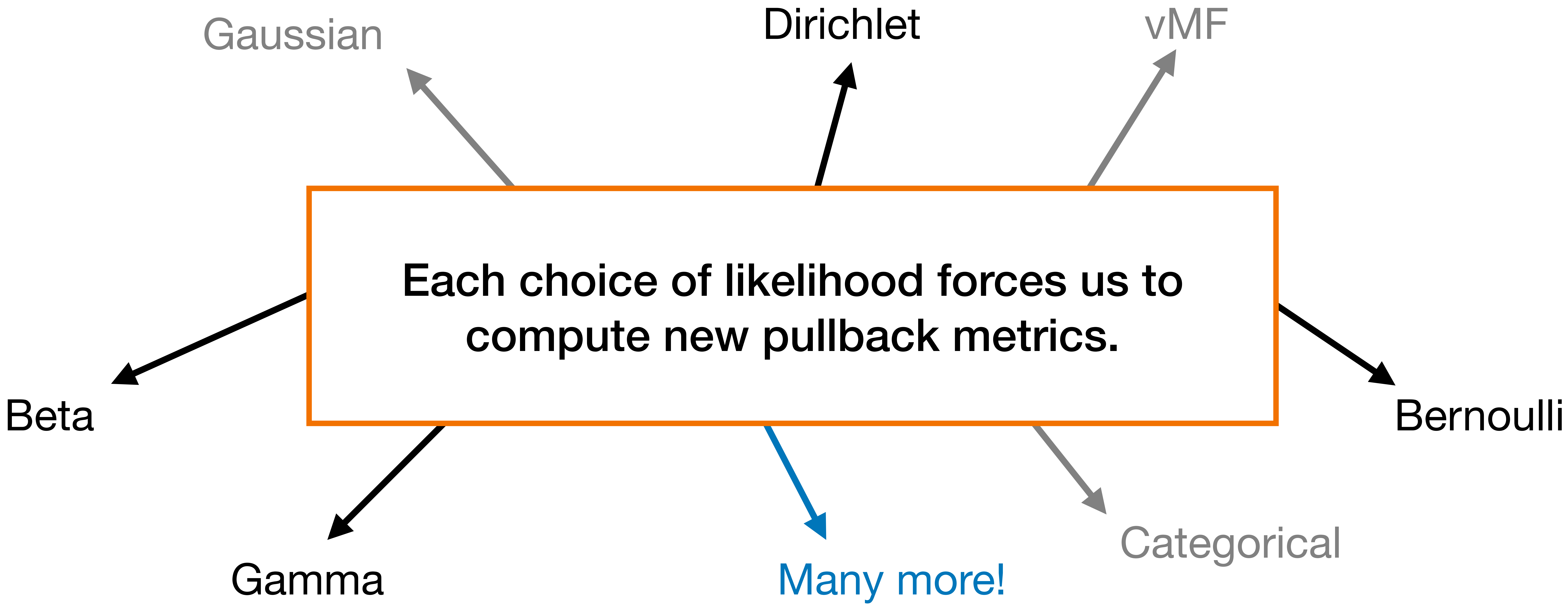
Each choice of likelihood forces us to compute new pullback metrics.

$$J^p_\mu(z)^\top J^p_\mu(z) + J^p_\sigma(z)^\top J^p_\sigma(z)$$
$$+ J^r_\mu(z)^\top J^r_\mu(z) + J^r_\kappa(z)^\top J^r_\kappa(z)$$

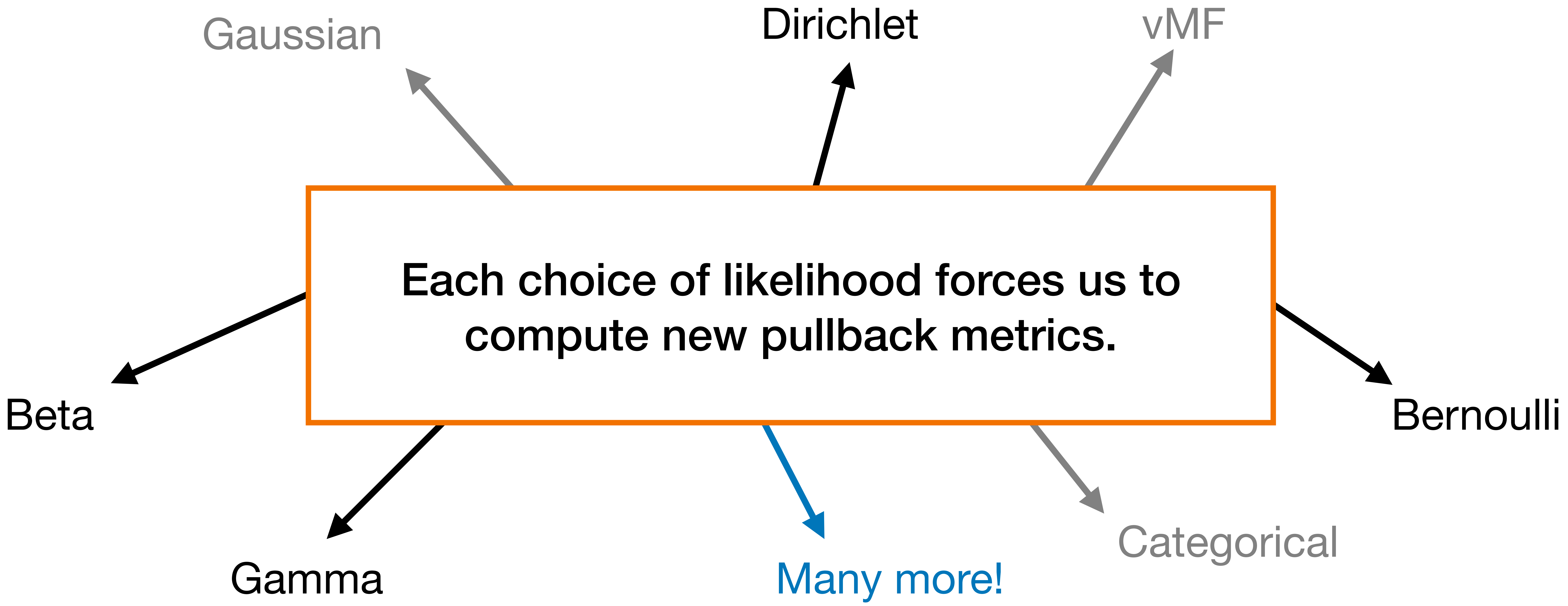
Pulling back information geometry


Gaussian
vMF
Each choice of likelihood forces us to compute new pullback metrics.
Categorical

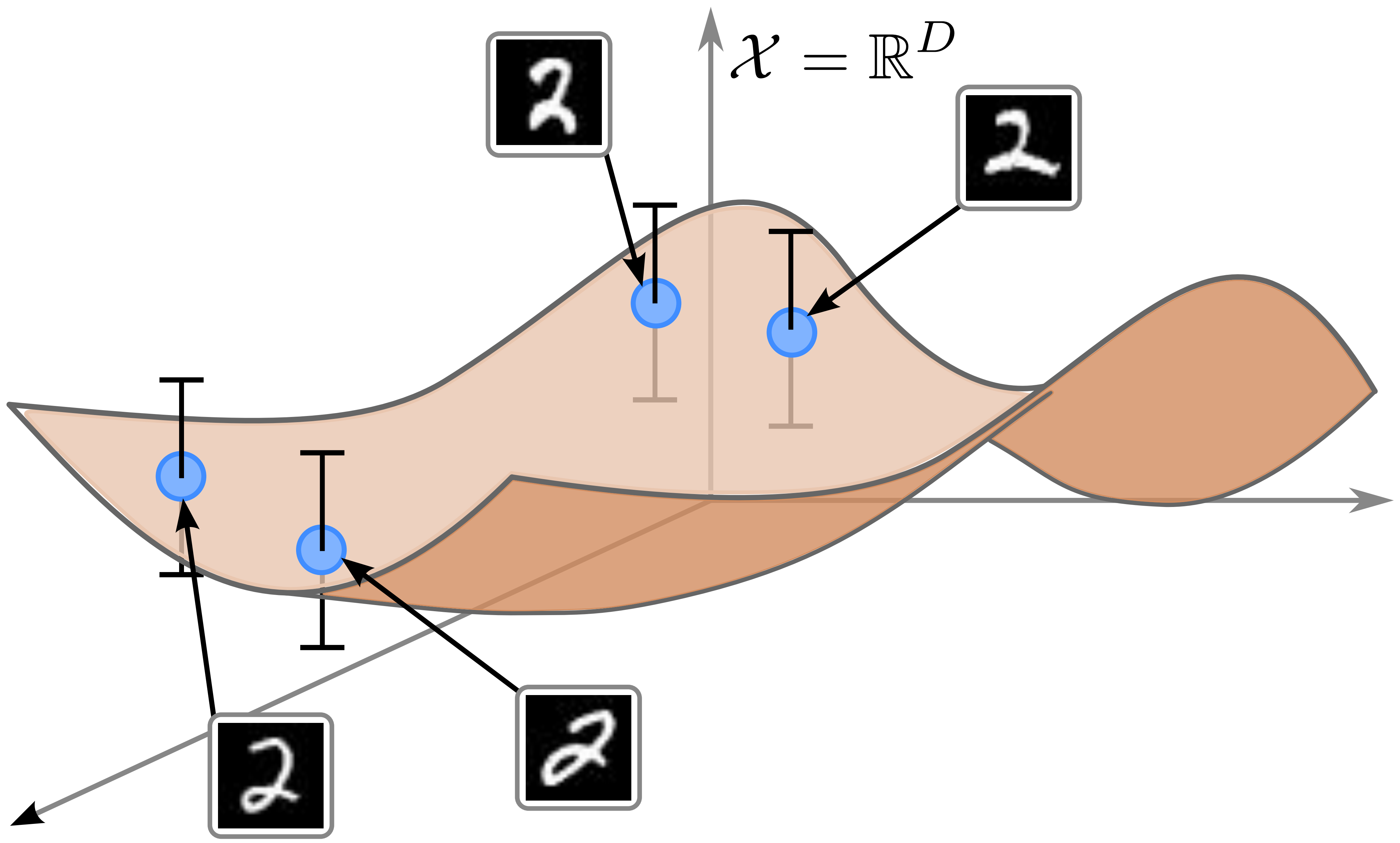
Before PBIG


Gaussian
Dirichlet
vMF
Each choice of likelihood forces us to compute new pullback metrics.
Beta
Bernoulli
Gamma
Many more!
Categorical

After PBIG

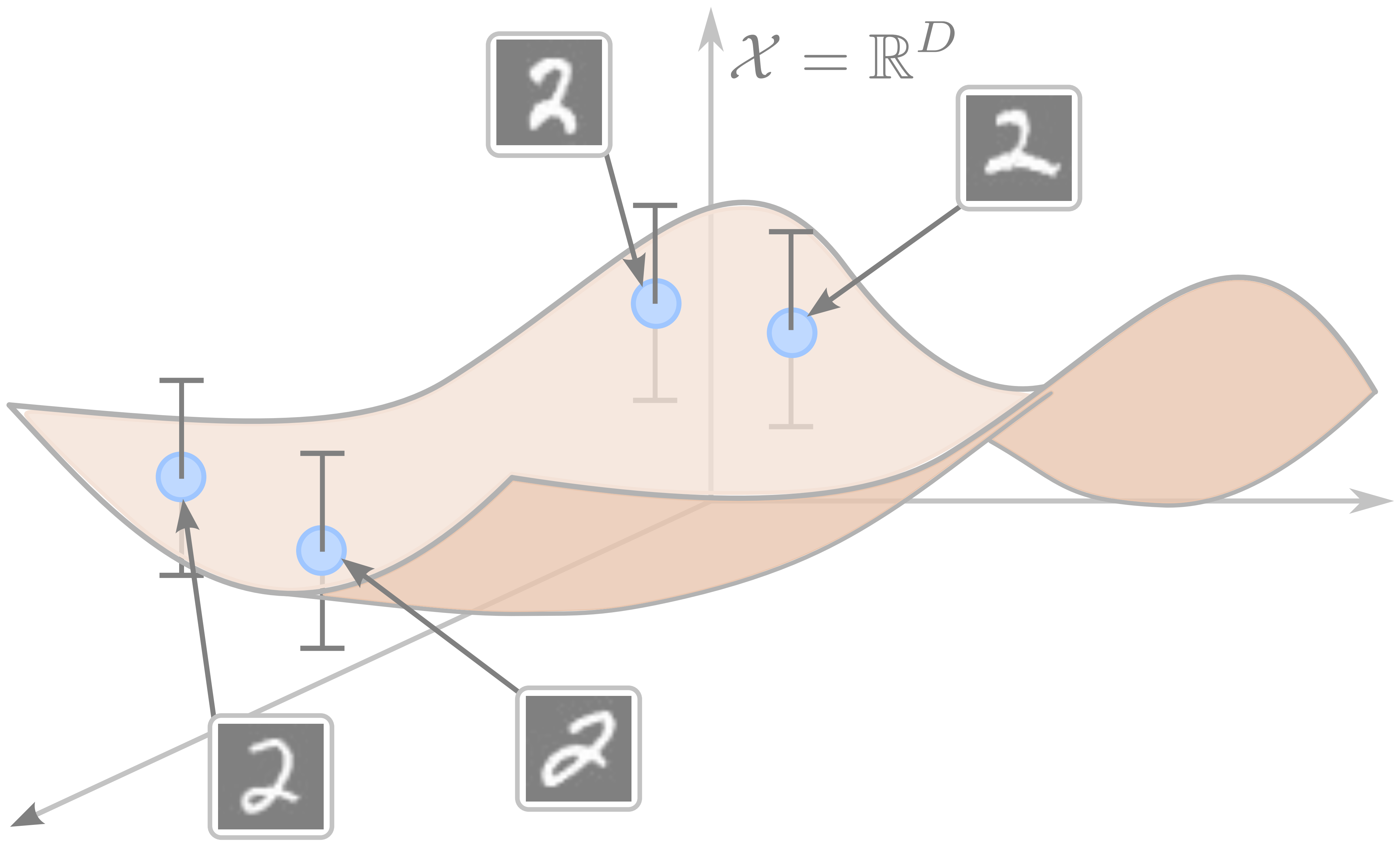


If you can sample from it differentially, you can get a latent space geometry!

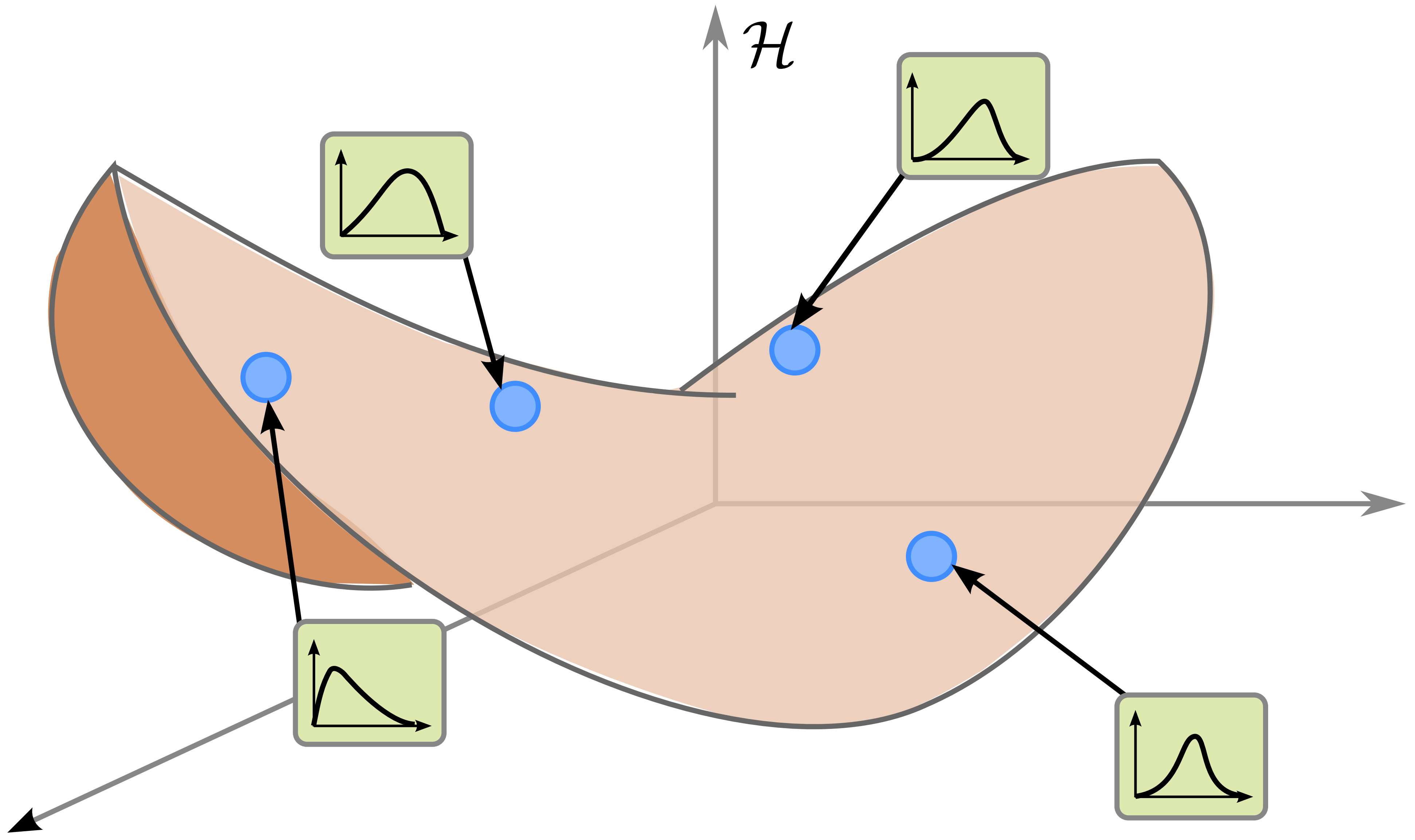


In all these applications, we decoded to data space.

How?



In all these applications, we decoded to data space.

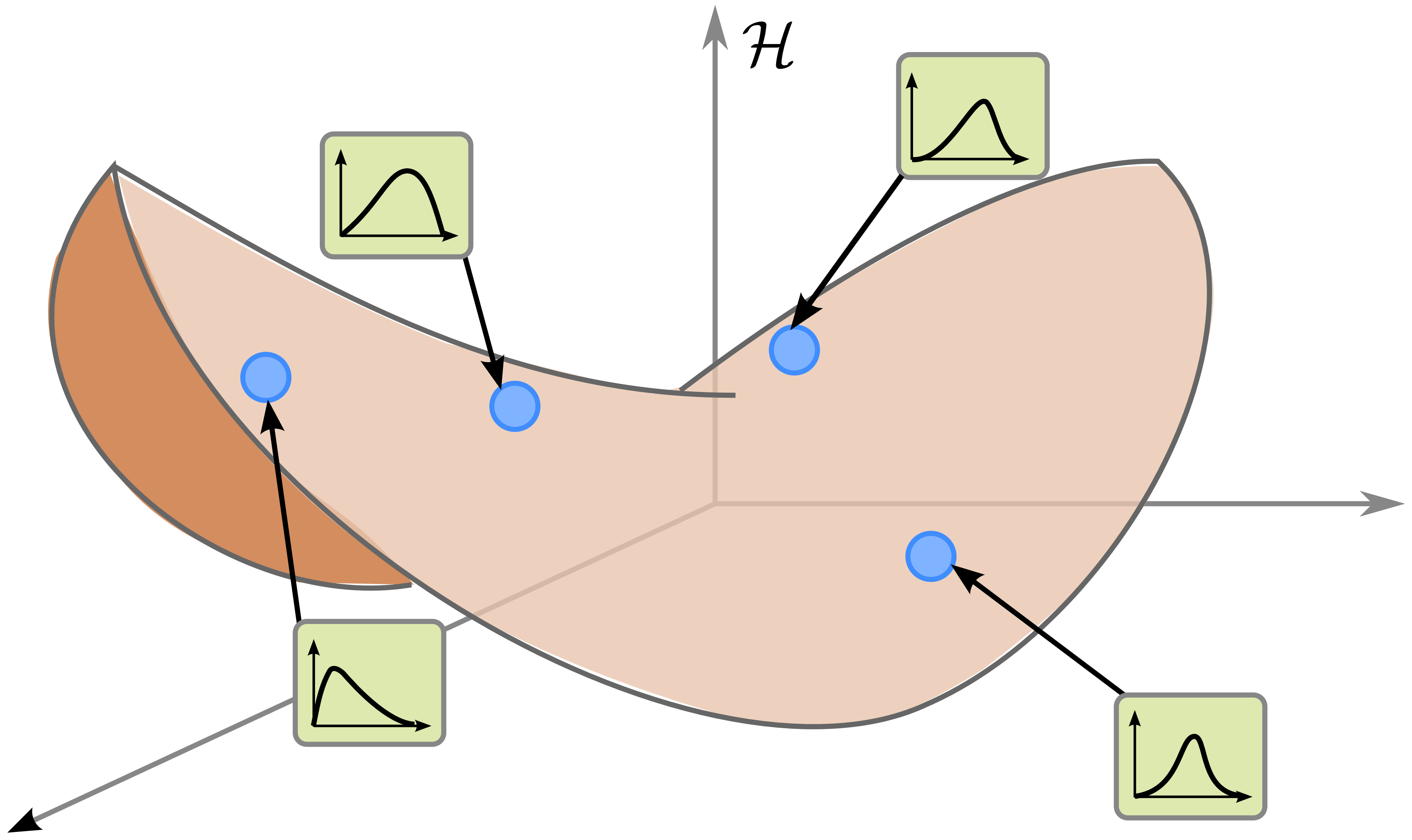


What if we decode to parameter space?

How?

Def. Given a distribution $p(x \mid \eta)$, we define its statistical manifold

$$(\mathscr{H}, I_{\mathscr{H}})$$



What if we decode to parameter space?

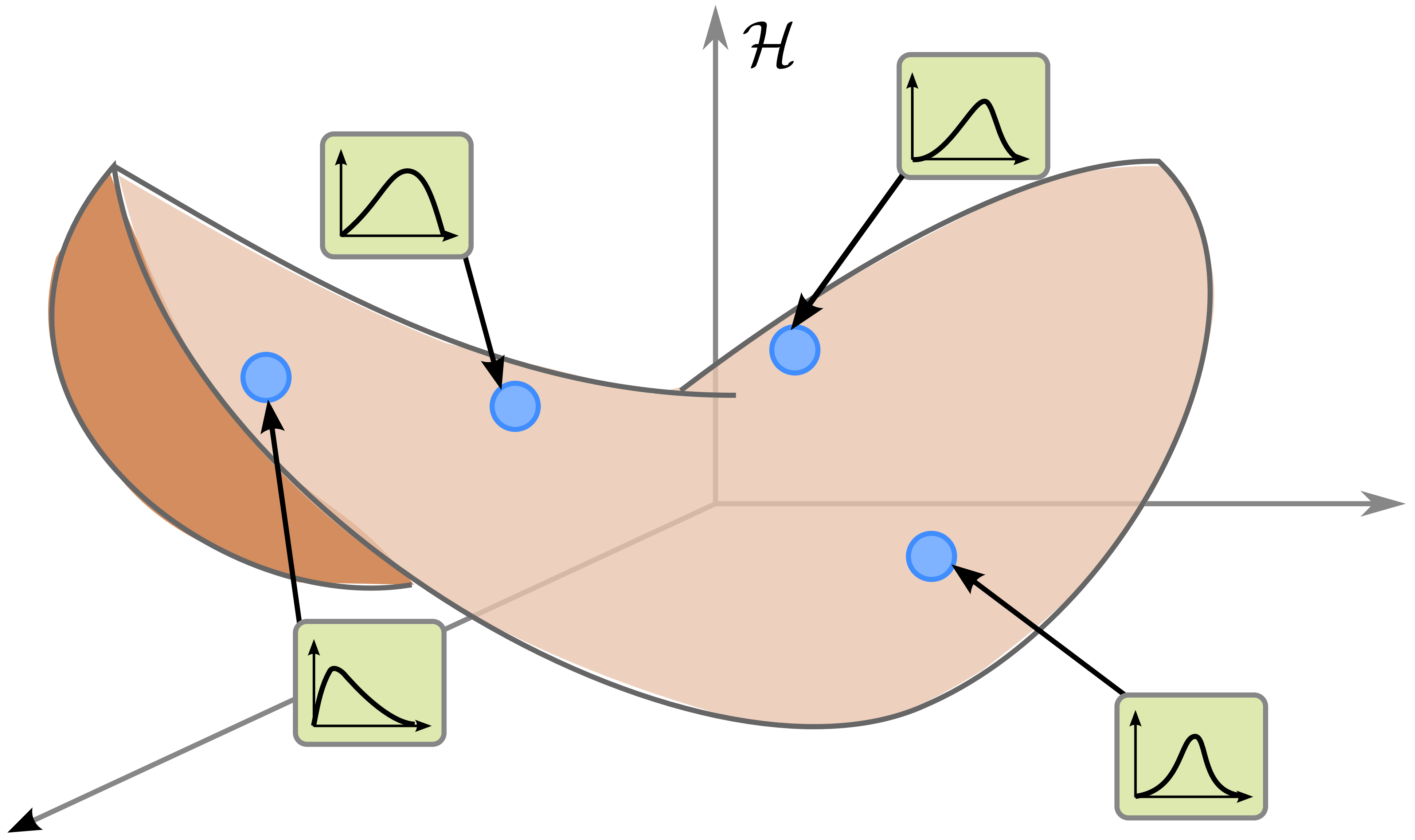
How?

Def. Given a distribution $p(x \mid \eta)$, we define its statistical manifold

$$(\mathscr{H}, I_{\mathscr{H}})$$

Set of parameters

Fisher Information Matrix (i.e. Fisher-Rao metric)



What if we decode to parameter space?

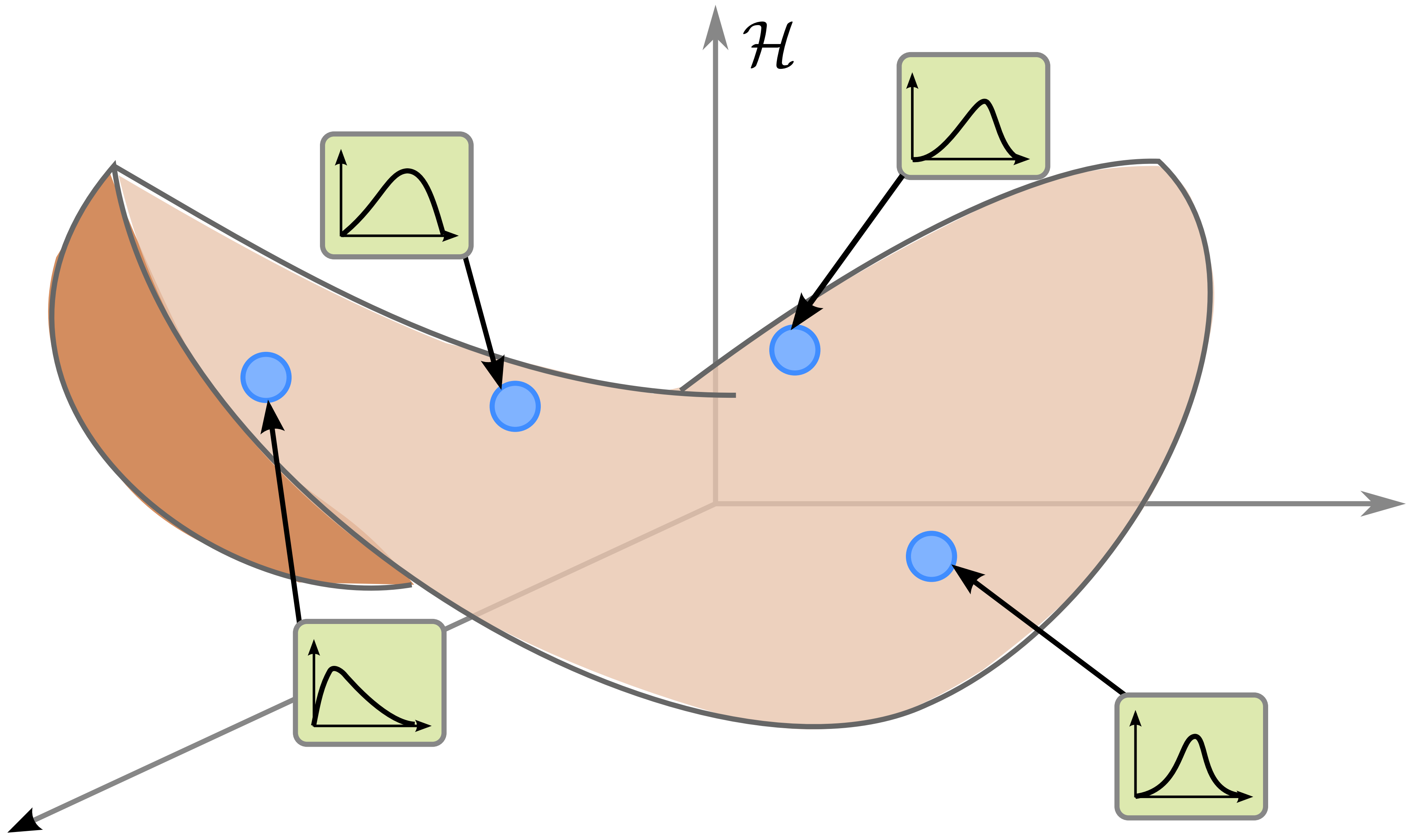
How?

Def. Given a distribution $p(x \mid \eta)$, we define its statistical manifold

$$(\mathscr{H}, I_{\mathscr{H}})$$

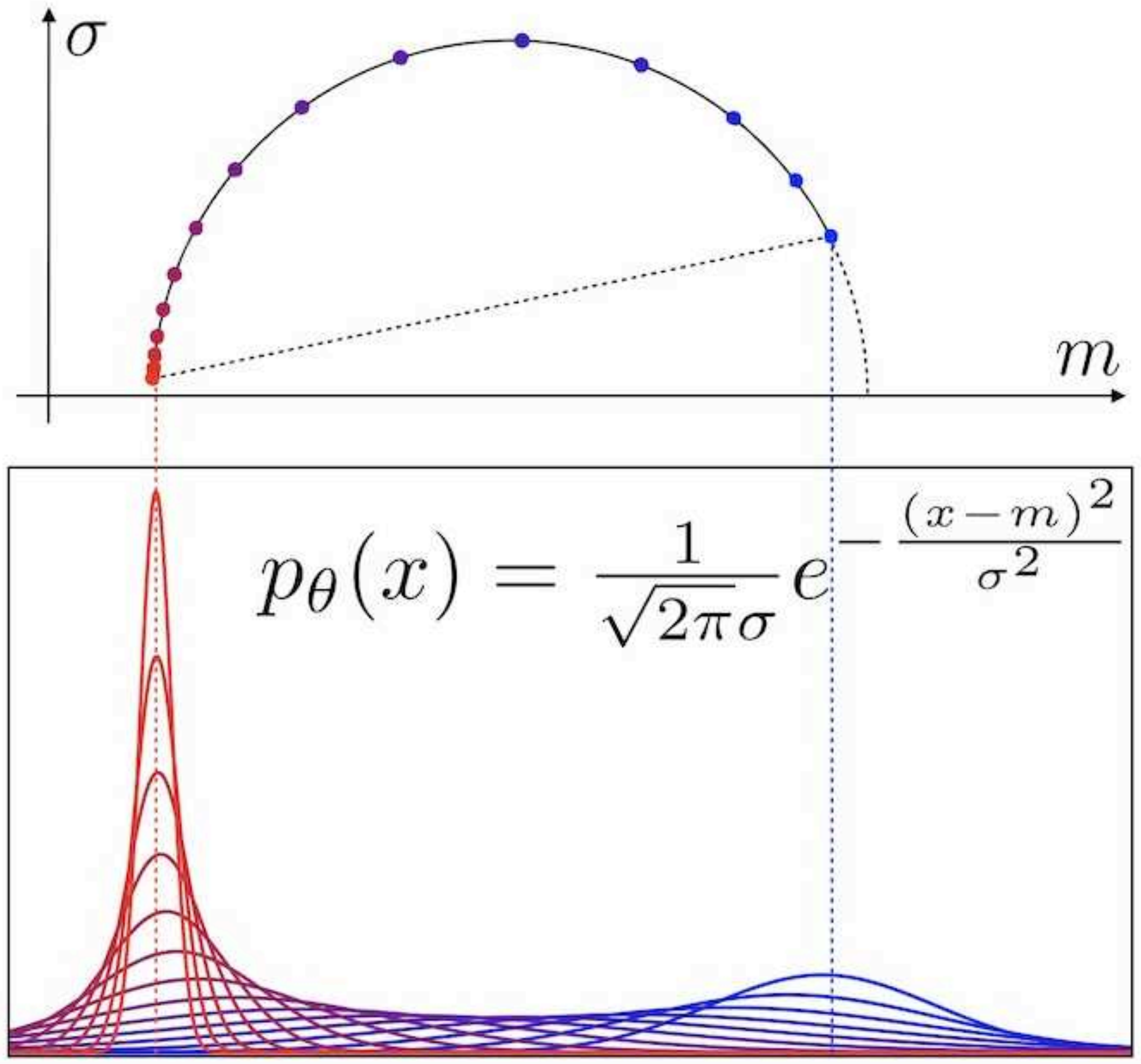
Set of parameters

Fisher Information Matrix
(i.e. Fisher-Rao metric)



What if we decode to parameter space?

$$I_{\mathscr{H}}(\eta) = \int_{\mathscr{X}} [\nabla_\eta \log p(x \mid \eta) \nabla_\eta \log p(x \mid \eta)^\top] p(x \mid \eta) \mathrm{d}x \,.$$

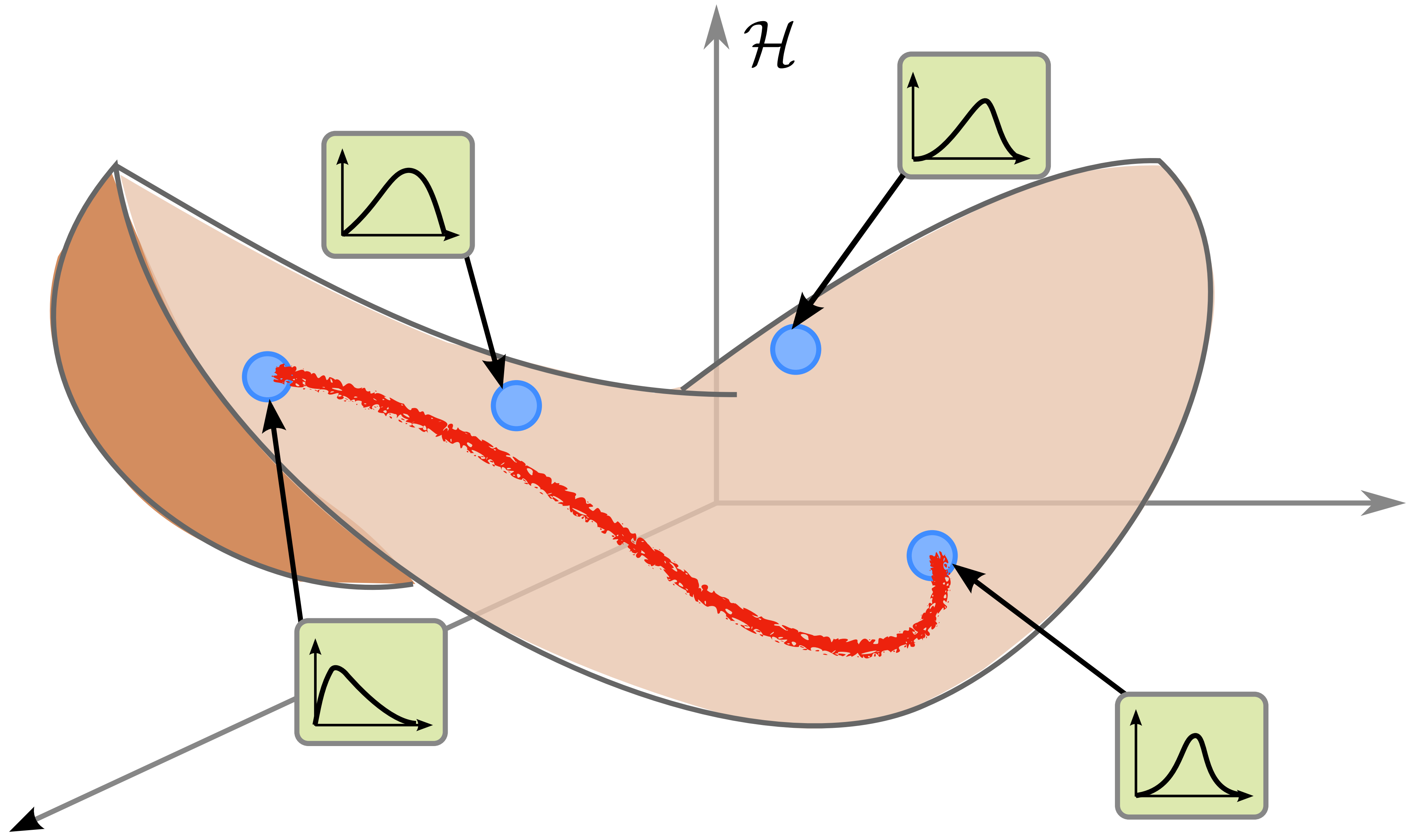


Gabriel Peyré
@gabrielpeyre

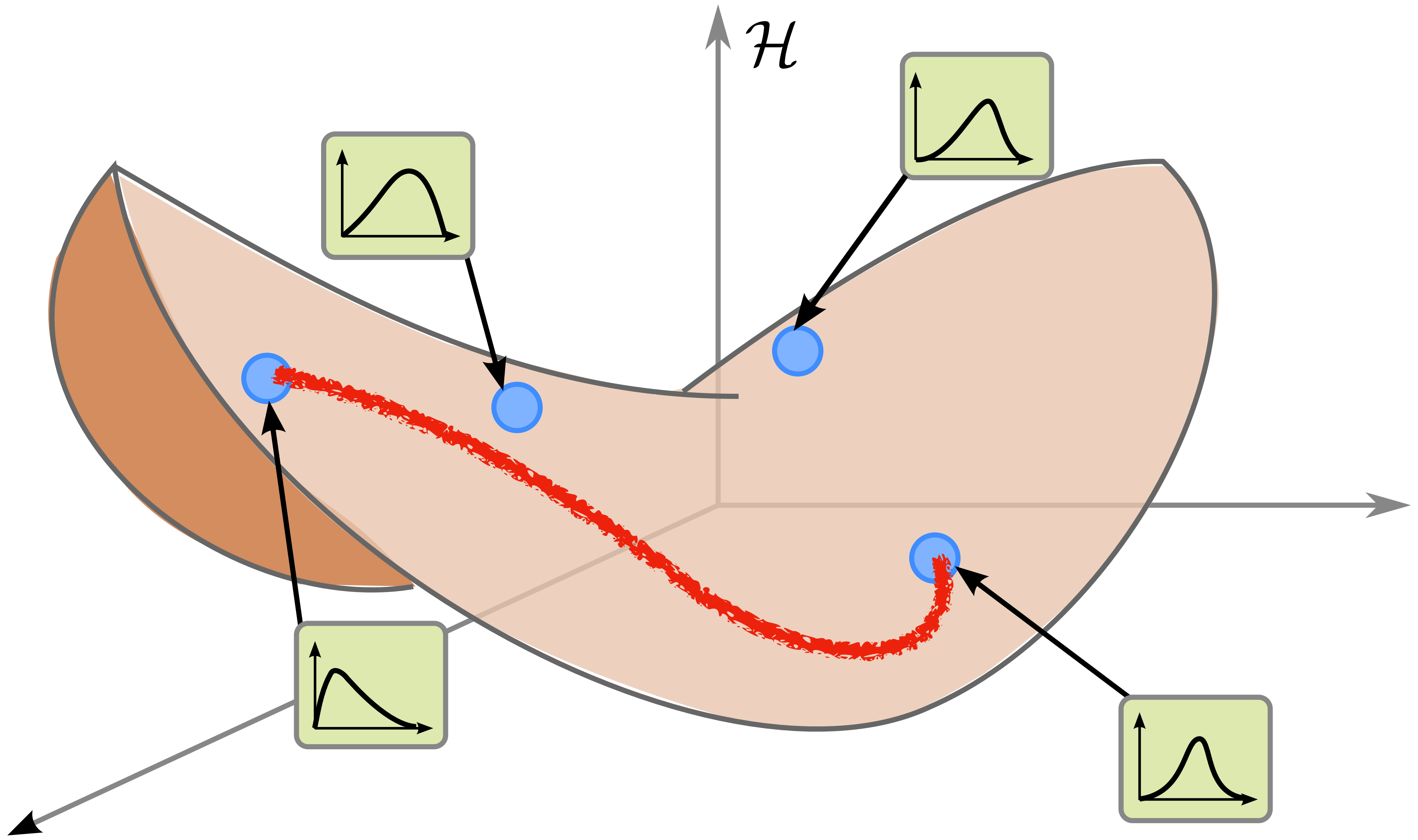
For the univariate Gaussian

$$\mathscr{H} = \{(\mu, \sigma) : \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$$

An example



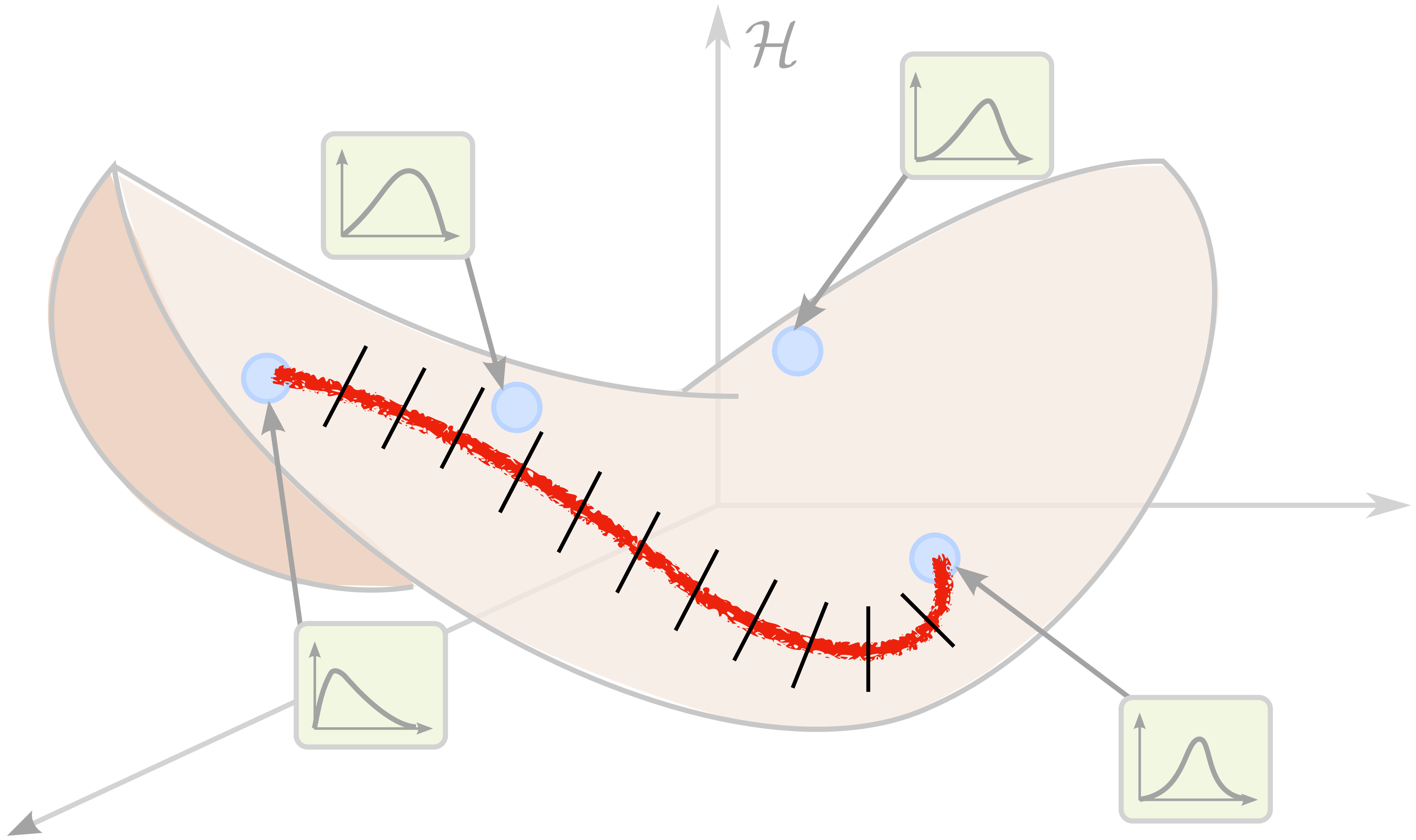
Q: How do we pull back the Fisher-Rao metric?



Q: How do we pull back the Fisher-Rao metric?

A: **Proposition 3.1.** *The Fisher-Rao metric is the second order approximation of the KL-divergence between perturbed distributions:*

$$\mathrm{KL}(p(\mathbf{x}|\eta), p(\mathbf{x}|\eta+\delta\eta)) = \frac{1}{2}\delta\eta^{\intercal}\mathbf{I}_{\mathcal{H}}(\eta)\delta\eta + o(\delta\eta^2). \quad (8)$$



Q: How do we pull back the Fisher-Rao metric?

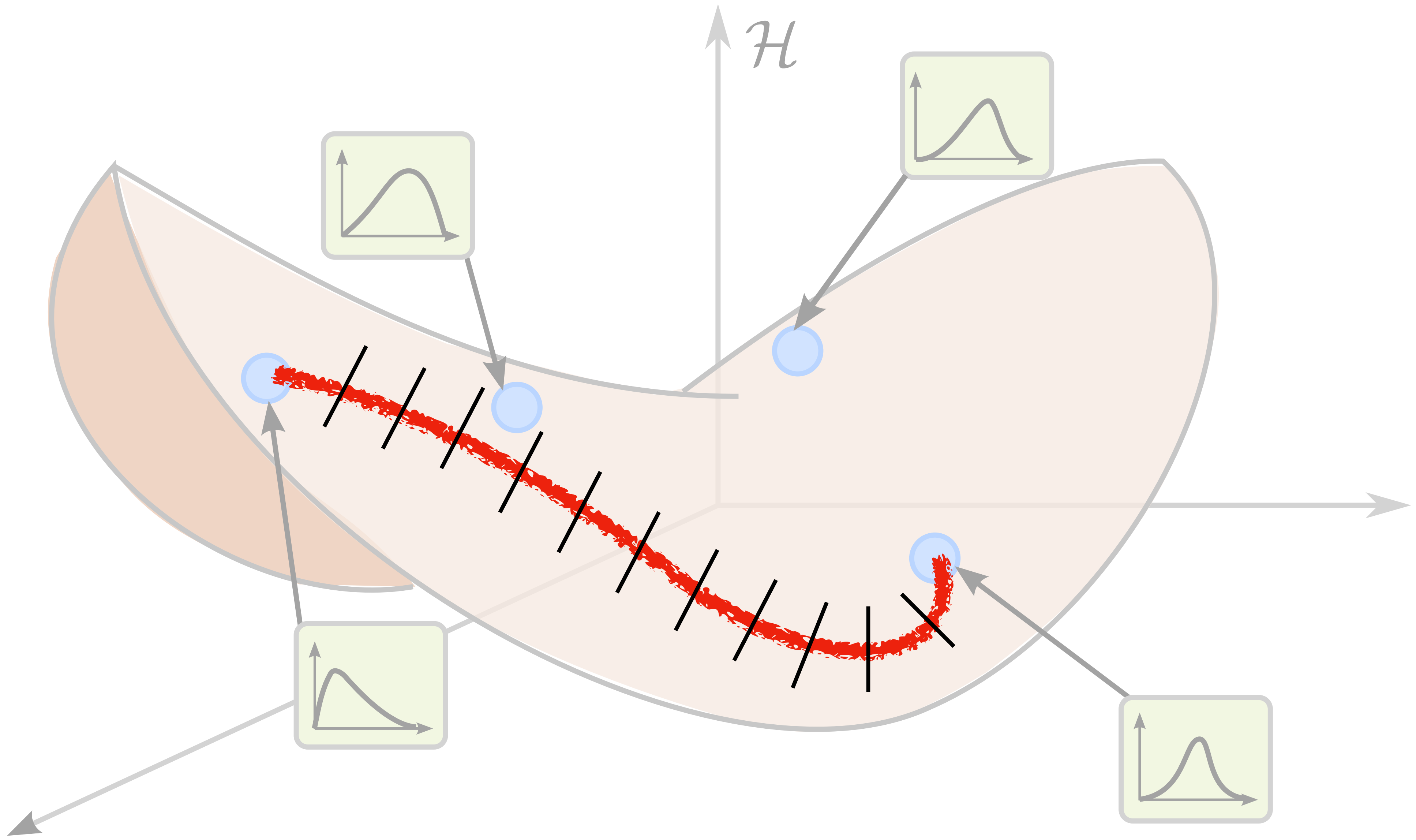
A: **Proposition 3.1.** *The Fisher-Rao metric is the second order approximation of the KL-divergence between perturbed distributions:*

$$\mathrm{KL}(p(\mathbf{x}|\eta), p(\mathbf{x}|\eta + \delta\eta)) = \frac{1}{2}\delta\eta^\intercal \mathbf{I}_{\mathcal{H}}(\eta)\delta\eta + o(\delta\eta^2). \quad (8)$$

Computing local KL divergences is enough!

$$\text{Energy}[c] \propto \lim_{N\to\infty} \sum_{n=1}^{N-1} \mathrm{KL}(p(x\,|\,c(t_n)), p(x\,|\,c(t_{n+1})))$$

Q: How do we pull back the Fisher-Rao metric?

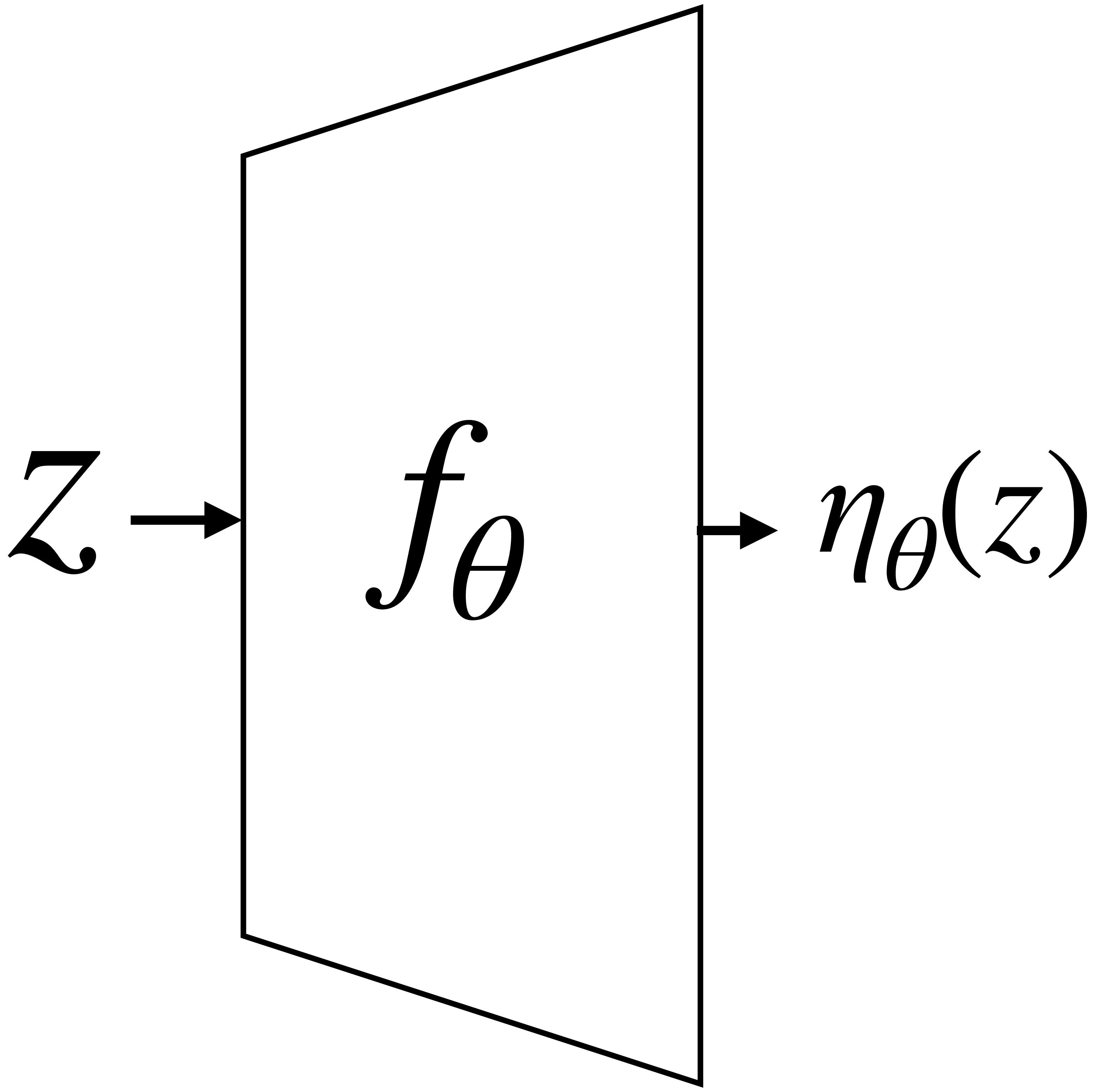


A: **Proposition 3.1.** *The Fisher-Rao metric is the second order approximation of the KL-divergence between perturbed distributions:*

$$\mathrm{KL}(p(\mathbf{x}|\eta), p(\mathbf{x}|\eta + \delta\eta)) = \frac{1}{2}\delta\eta^{\intercal}\mathbf{I}_{\mathcal{H}}(\eta)\delta\eta + o(\delta\eta^2). \quad (8)$$

Computing local KL divergences is enough!

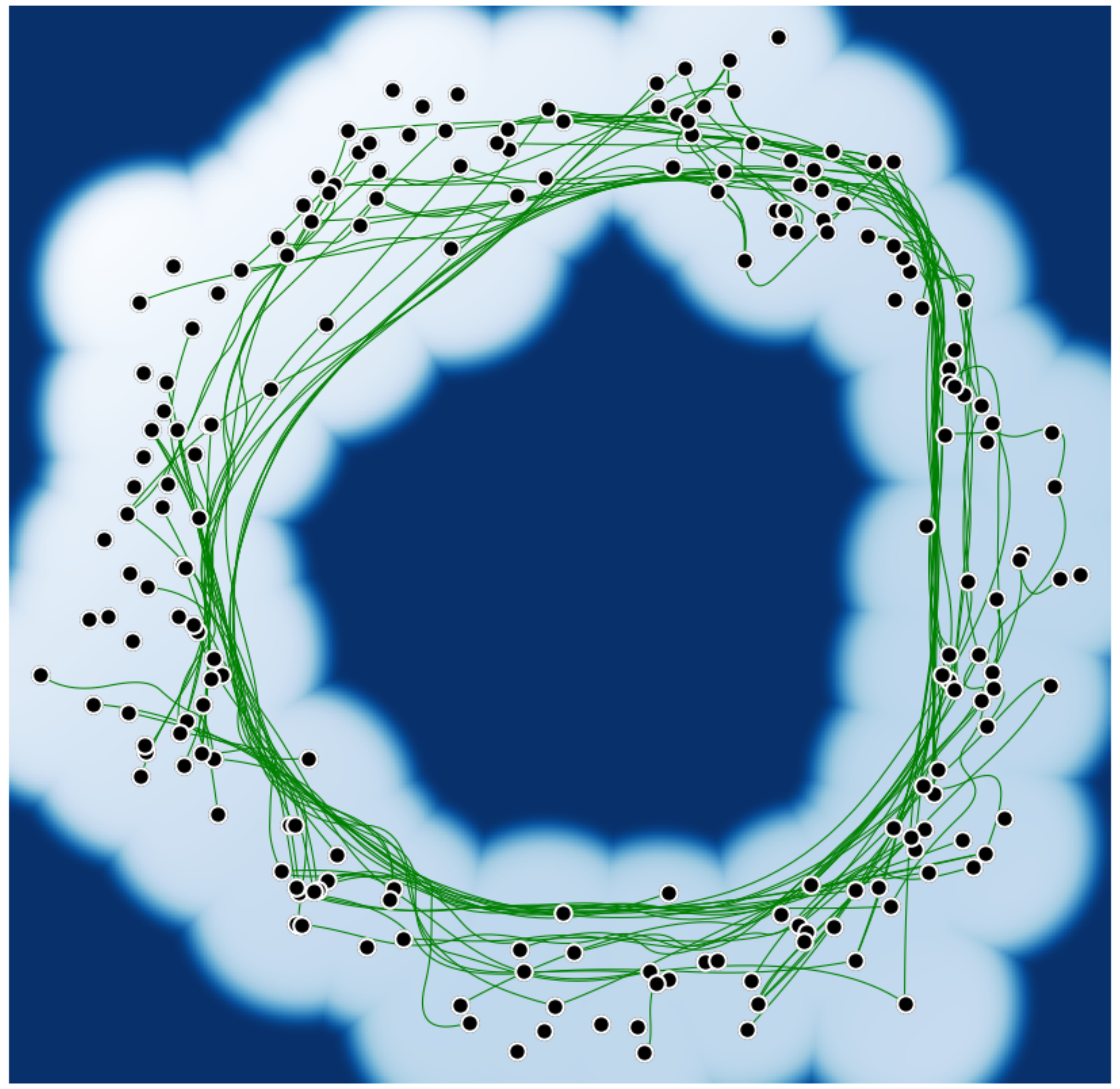
Easily minimized using e.g. `torch` ⟵ $\text{Energy}[c] \propto \lim_{N\to\infty} \sum_{n=1}^{N-1} \mathrm{KL}(p(x\,|\,c(t_n)), p(x\,|\,c(t_{n+1})))$



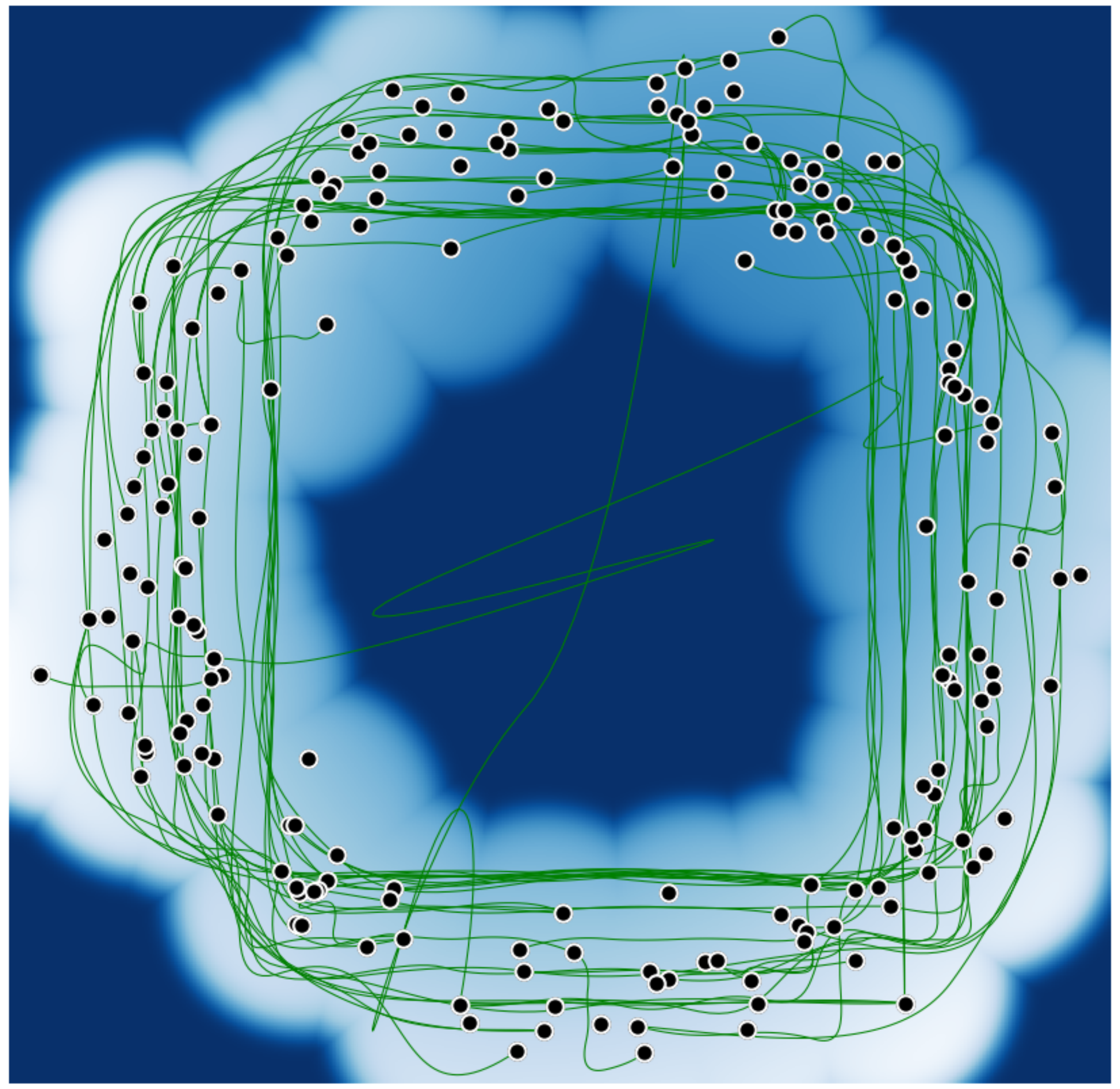
We pull back the Fisher-Rao for

- Normal
- Bernoulli
- Beta
- Dirichlet
- Exponential

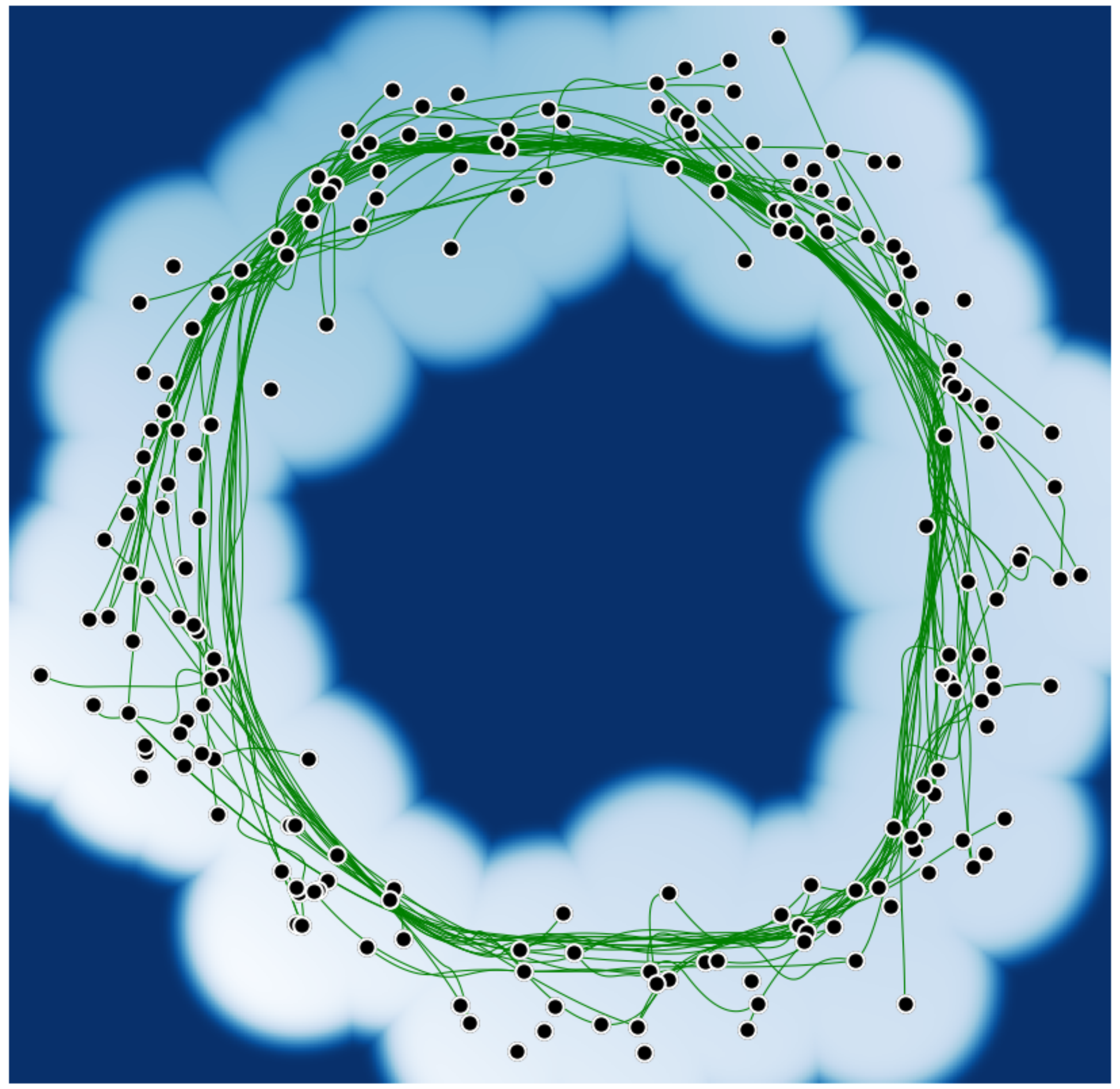
A toy experiment



Beta

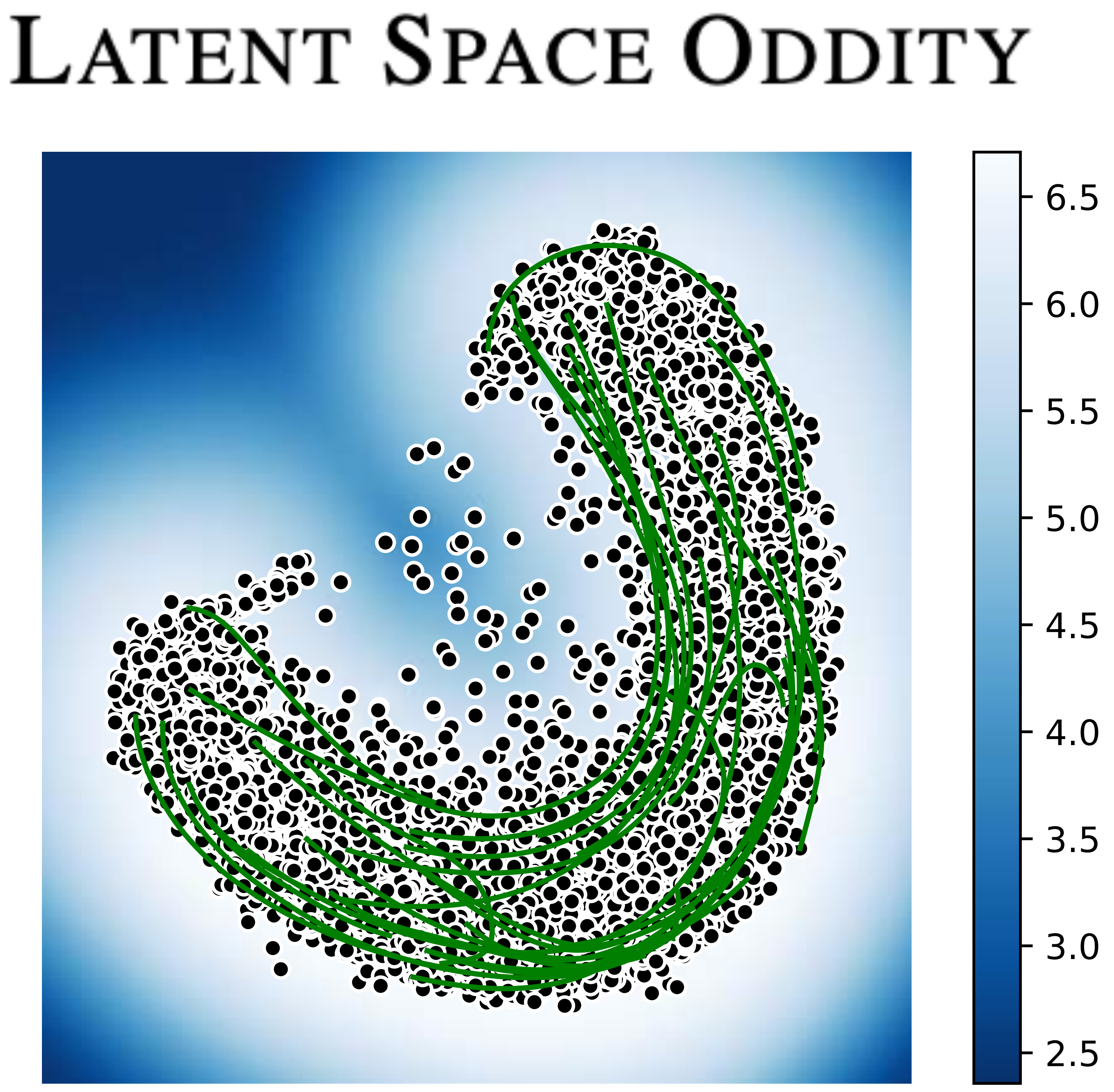


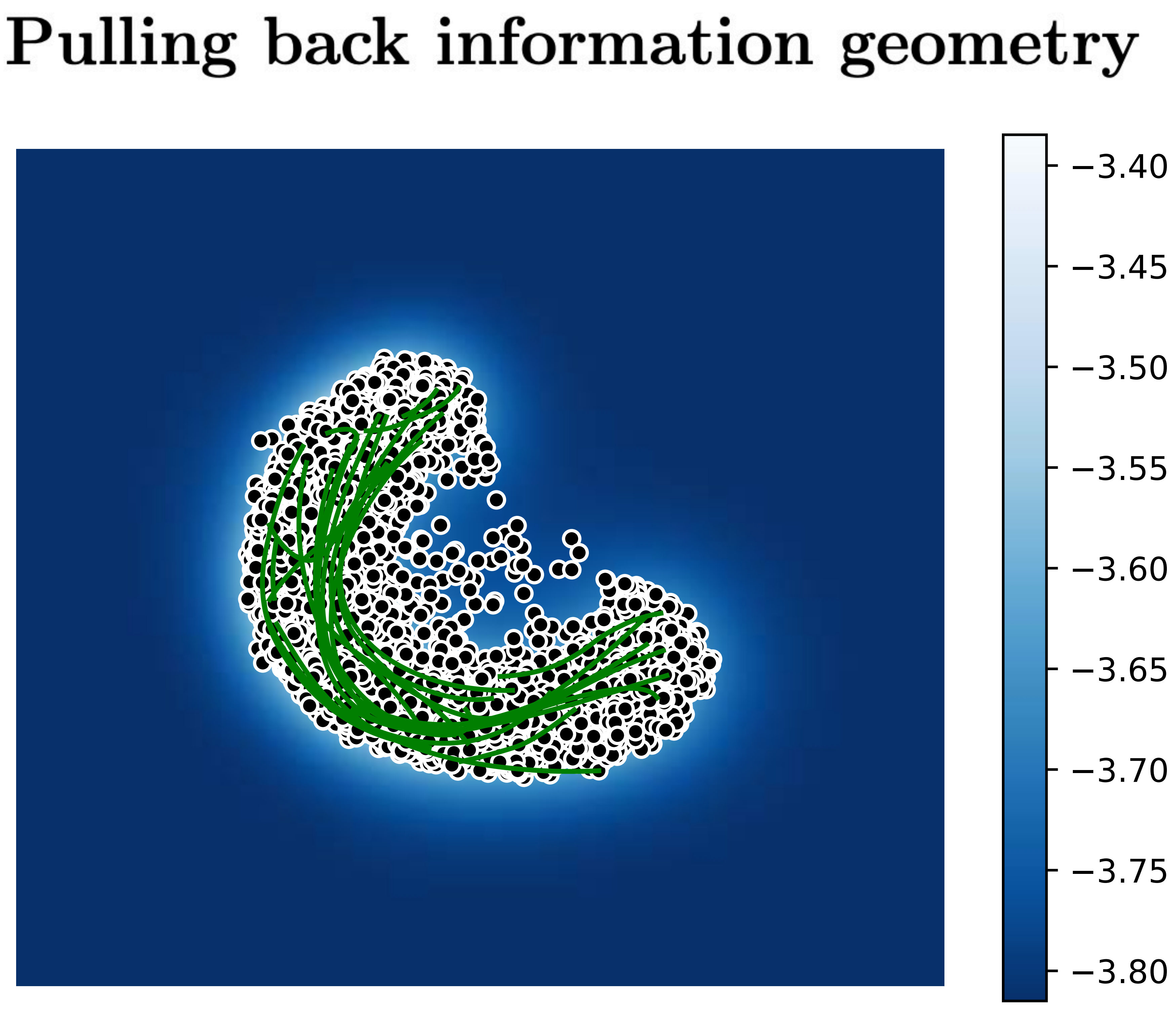
Bernoulli



Dirichlet

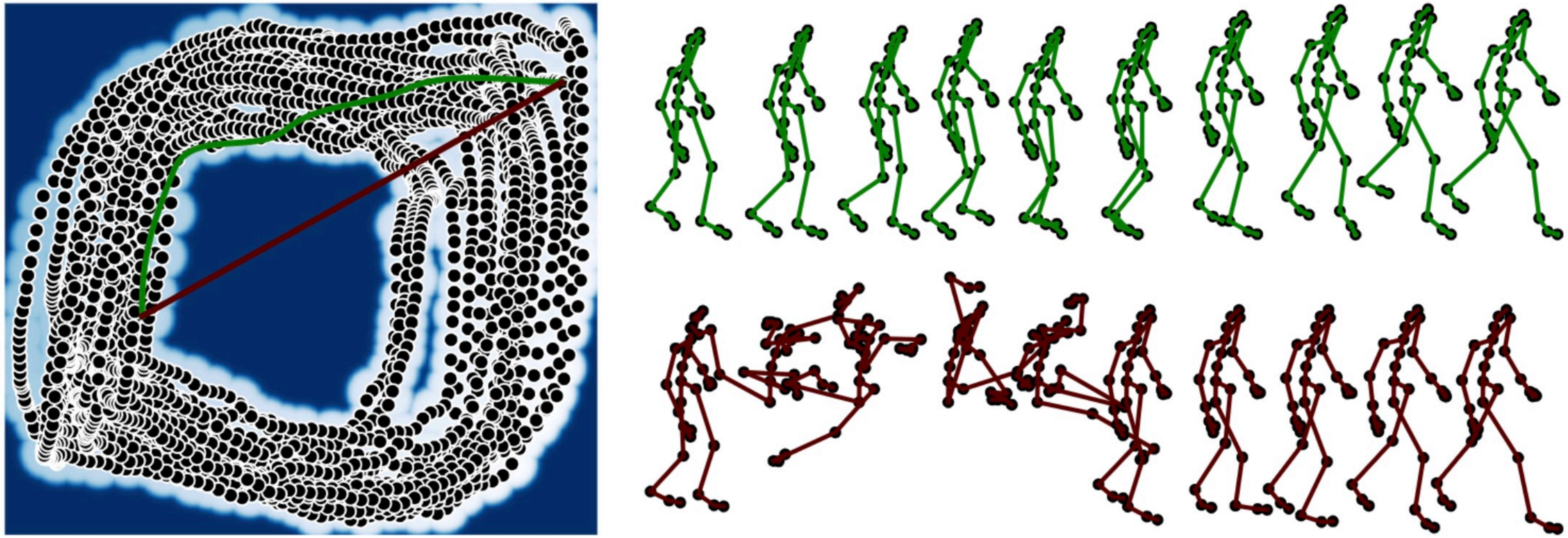
A toy experiment





Decoding to a Gaussian on MNIST(1)

Comparing against Latent Space Oddity

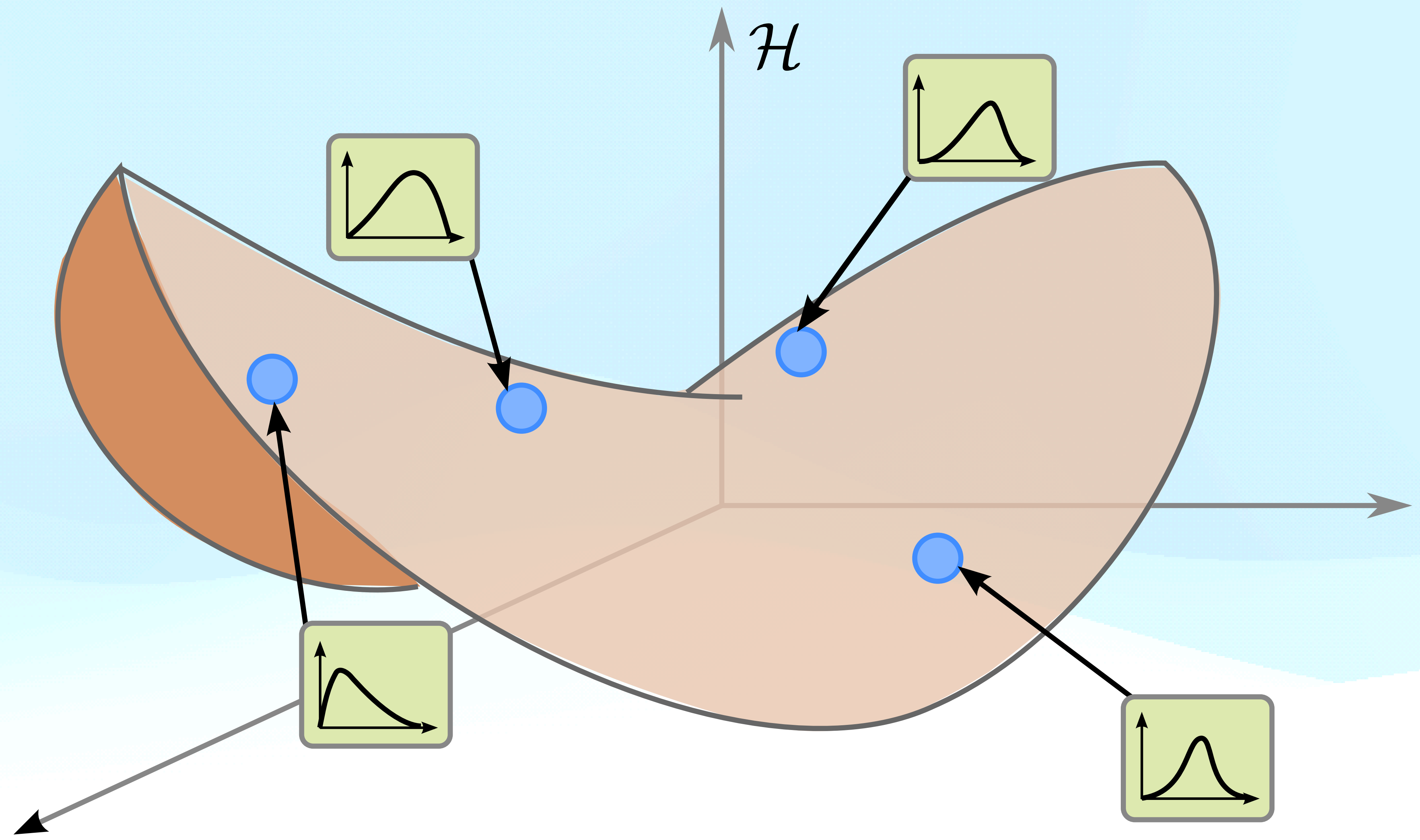


Human poses live on $\mathbb{T}^n = \mathbb{S}^1 \times \cdots \times \mathbb{S}^1$ (product of vMF)

On human poses

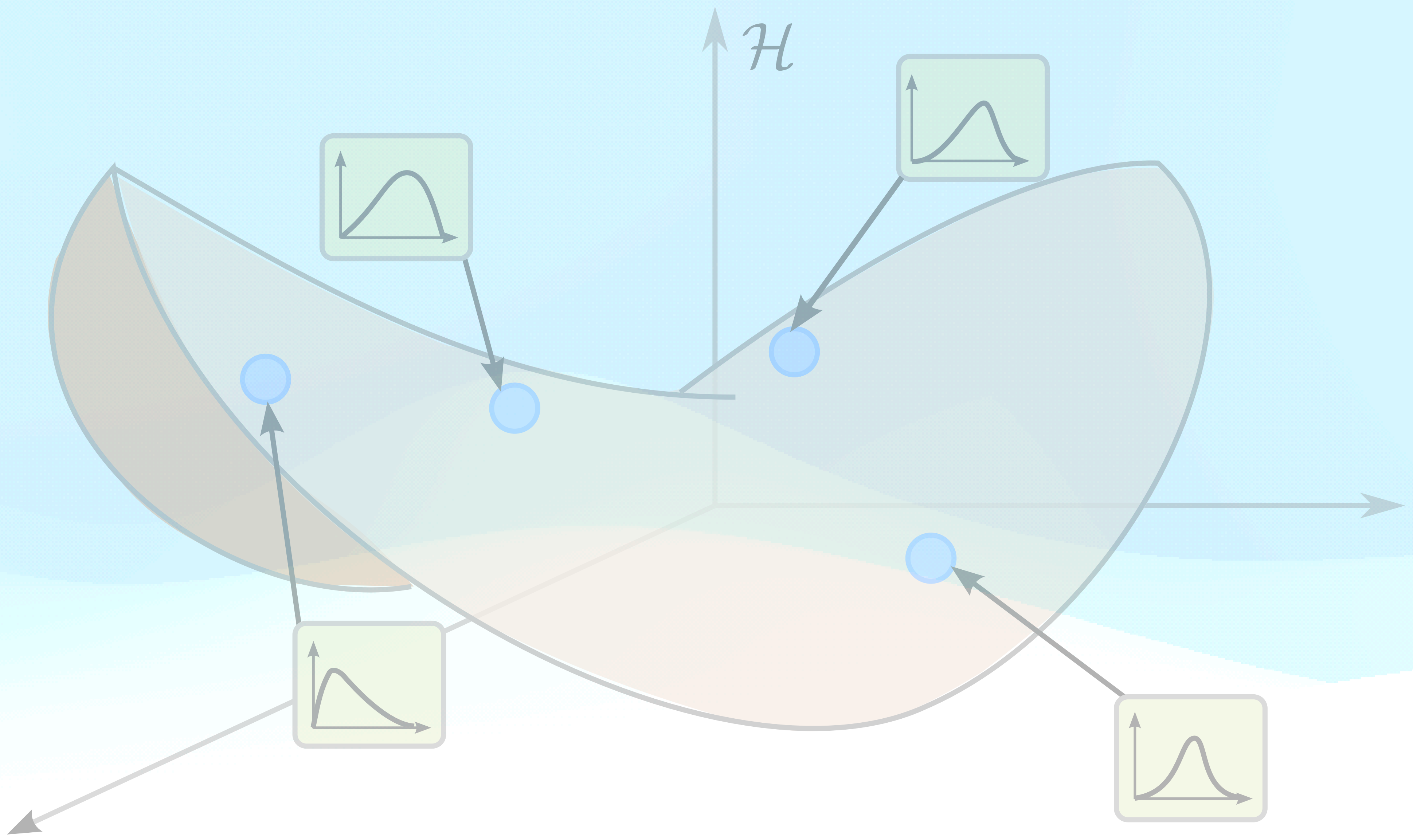
Summary

We consider parameter space instead of data space



Summary

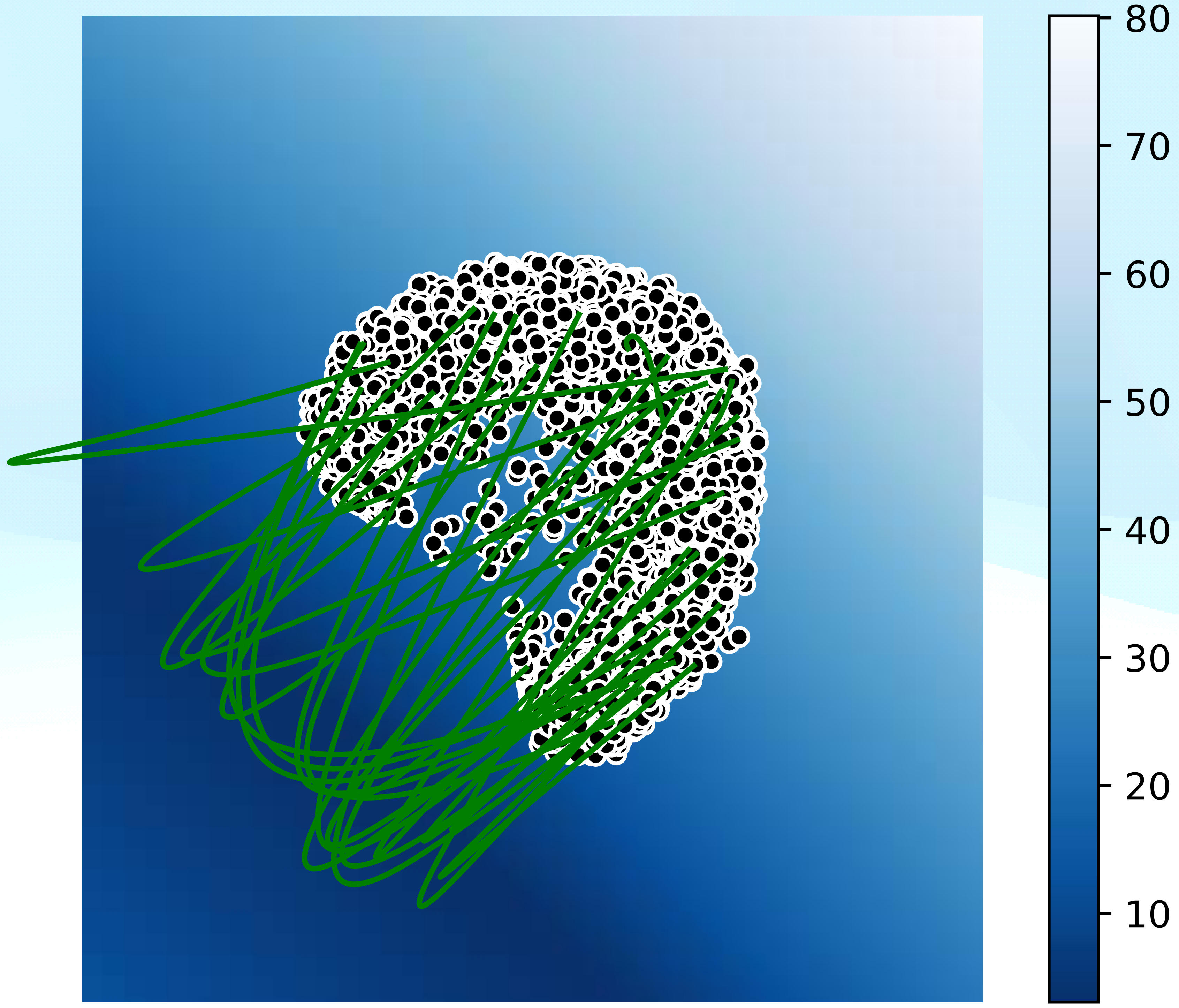
We consider parameter space instead of data space



This allows us to define *black box latent geometries*

$$\sum_{n=1}^{N-1} \mathrm{KL}(p(x \mid c(t_n)), p(x \mid c(t_{n+1})))$$

Good uncertainty quantification is vital for latent geometries

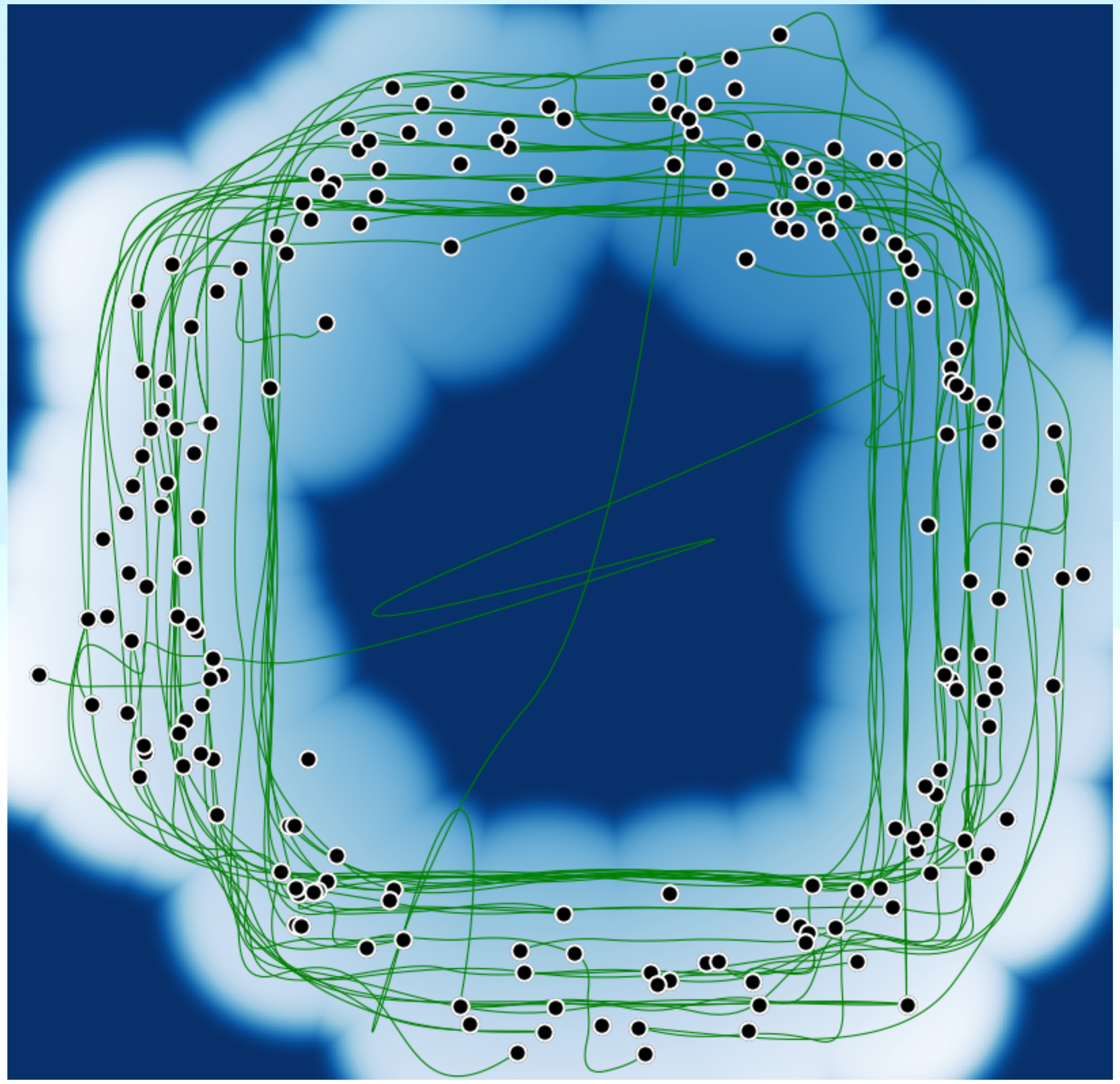


$$\widetilde{\sigma}_\theta(z) = \begin{cases} \sigma_\theta(z) \\ \text{a large number} \end{cases}$$

Outlook - open problems

Good uncertainty quantification is vital for latent geometries

What does uncertain mean in other distributions?



Outlook - open problems

Thanks! Any questions?