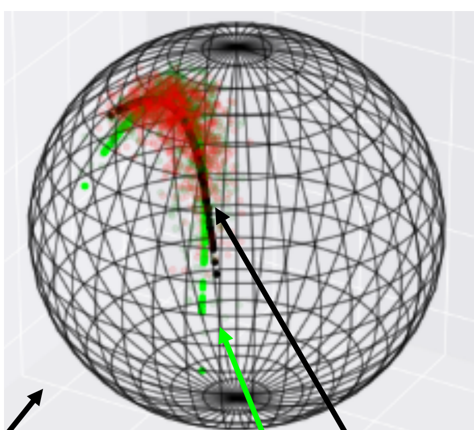
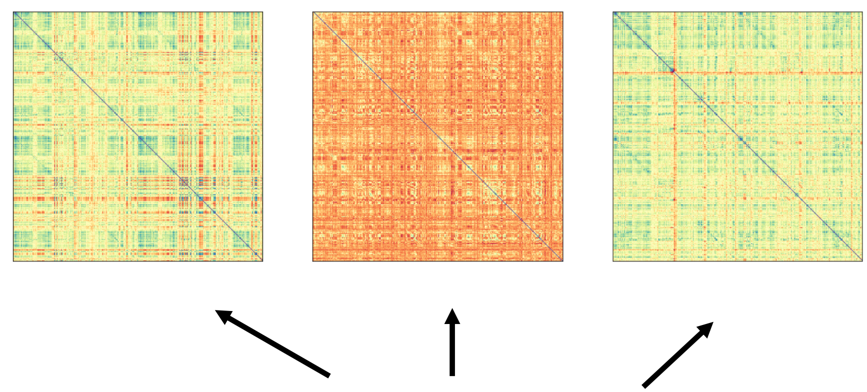
Learning submanifolds with geometric variational autoencoders:

Application to brain functional connectomes

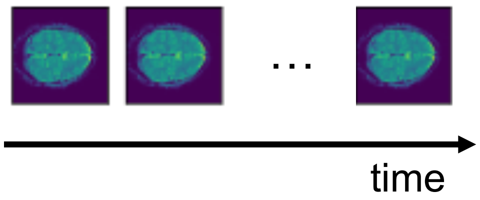
Nina Miolane, Postdoctoral Fellow and Lecturer @ Stanford Statistics, Holmes Lab





Data from Biomedical Images

- Datasets sizes n are *relatively* small & number of parameters p is *relatively* large, *compared to* traditional computer vision image datasets.



Usual functional MRI dataset:
- n = 400 “3D videos”

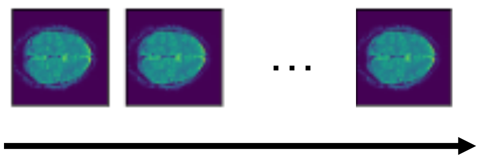
“3D videos”, size:
- p = 256x256x192x4800 = 60B

MNIST dataset:
- n = 70k images

2D images, size:
- p = 28x28 = 784

Data from Biomedical Images

- Datasets sizes n are *relatively* small & number of parameters p is *relatively* large, *compared to* traditional computer vision image datasets.



Usual functional MRI dataset:
- n = 400 "3D videos"

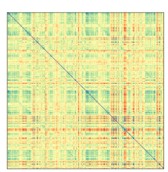
"3D videos", size:
- p = 256x256x192x4800 = 60B

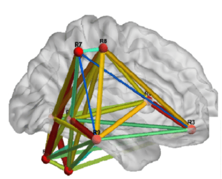
MNIST dataset:
- n = 70k images

2D images, size:
- p = 28x28 = 784

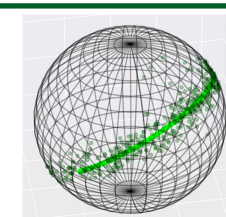
→ **Data preprocessing:** Extract meaningful features from images: data are represented as elements of manifolds, i. e. of non-linear spaces.

→ **Data analysis:** Use relatively non-flexible statistical models: eg. equivalent of PCA.





Functional connectome
$\in$ manifold $\mathrm{SPD}(n_{\mathrm{nodes}})$
of dimension $n_{\mathrm{nodes}}\,(n_{\mathrm{nodes}}+1)\,/\,2$
Ex. for $n_{\mathrm{nodes}} = 15$: p = 120



Note: Riemannian geometry on the SPD manifold usually improves performances.

Data from Biomedical Images

- **Datasets sizes n are growing**

HUMAN Connectome PROJECT
- n = 1200 “3D videos”
- p = 60B

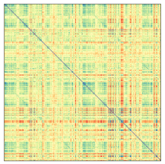
biobank uk
- n = 20k “3D videos”
- p = 60B

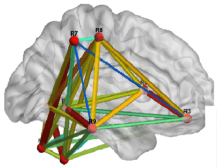
MNIST dataset:
- n = 70k images

2D images, size:
- p = 28x28 = 784

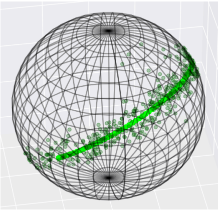
→ **Data preprocessing:** Extract meaningful features from images: data are represented as elements of manifolds, i. e. of non-linear spaces.

→ **Data analysis:** Use relatively non-flexible statistical models: eg. equivalent of PCA.





Functional connectome
$\in$ manifold $SPD(n_{nodes})$
of dimension $n_{nodes}\,(n_{nodes}+1)\,/\,2$
Ex. for n_{nodes} = 15: p = 120



Note: Riemannian geometry on the SPD manifold usually improves performances.

Data from Biomedical Images

- **Datasets sizes n are growing**

HUMAN Connectome PROJECT
- n = 1200 “3D videos”
- p = 60B

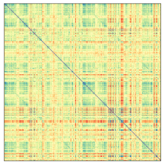
biobank uk
- n = 20k “3D videos”
- p = 60B

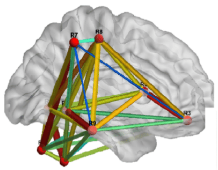
MNIST dataset:
- n = 70k images

2D images, size:
- p = 28x28 = 784

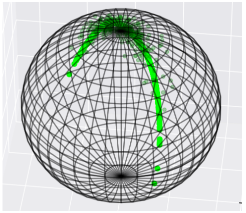
→ **Data preprocessing:** Extract meaningful features from images: data are represented as elements of manifolds, i. e. of non-linear spaces.

→ **Data analysis:** **Enable the use of more flexible statistical models.**





Functional connectome
$\in$ manifold $\mathrm{SPD}(n_{\mathrm{nodes}})$
of dimension $n_{\mathrm{nodes}}\,(n_{\mathrm{nodes}}+1)\,/\,2$
Ex. for n_{nodes} = 15: p = 120



Note: Riemannian geometry on the SPD manifold usually improves performances.

Statistical Models for Dimension Reduction

Assumption: data (approximately) lies on a low-dimensional space of the ambient space

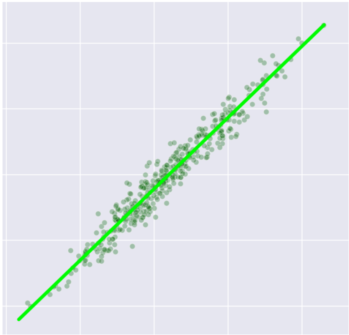
Ambient space: Euclidean $\mathbb{R}^D$

Ambient space: Manifold M

Less flexible model

Goal: Learn linear subspace…

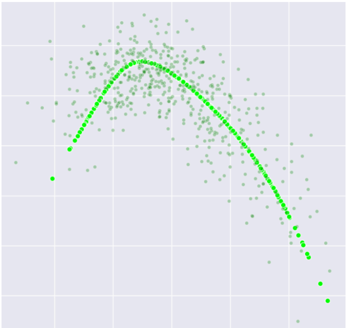
$\mathbb{R}^D = \mathbb{R}^2$:



More flexible model

Goal: Learn non-linear subspace…

$\mathbb{R}^D = \mathbb{R}^2$:



… and (approximate) posterior distributions for the low-dimensional representation of each data point.

Statistical Models for Dimension Reduction

Assumption: data (approximately) lies on a low-dimensional space of the ambient space

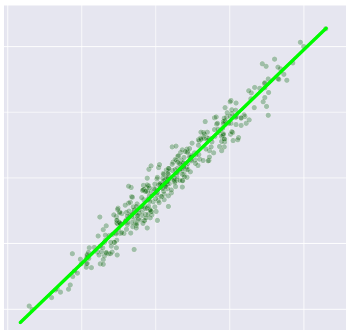
Ambient space: Euclidean $\mathbb{R}^D$

Ambient space: Manifold M

Less flexible model

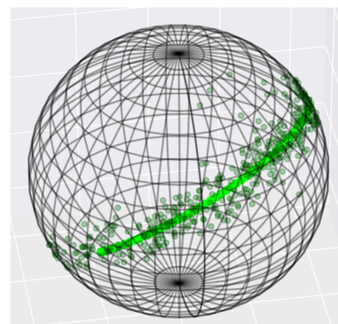
Goal: Learn linear subspace…

$\mathbb{R}^D = \mathbb{R}^2$:

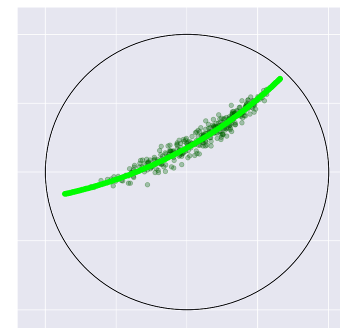


Goal: Learn "geodesic" submanifold…

$M = S^2$:



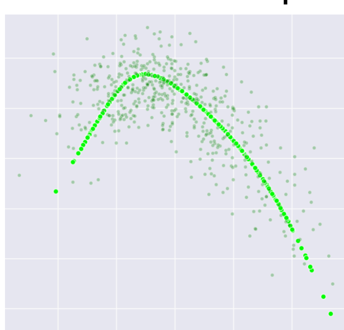
$M = H^2$:



More flexible model

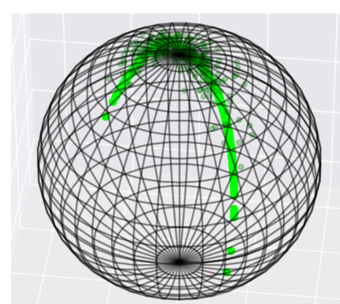
Goal: Learn non-linear subspace..

$\mathbb{R}^D = \mathbb{R}^2$:

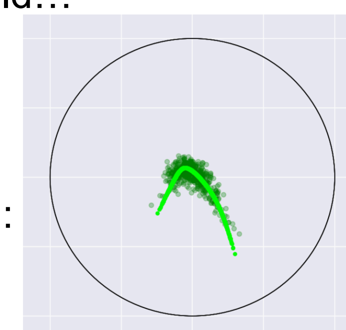


Goal: Learn "non-geodesic" submanifold…

$M = S^2$:



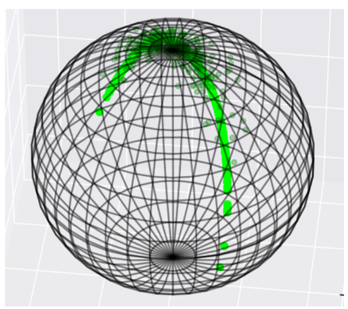
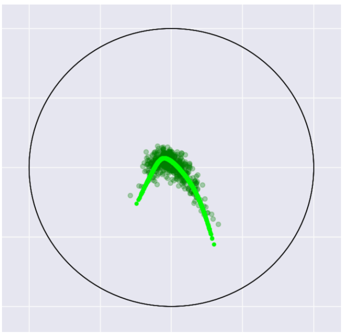
$M = H^2$:



… and (approximate) posterior distributions for the low-dimensional representation of each data point.

Statistical Models for Dimension Reduction

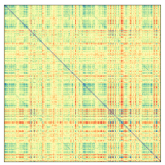
Assumption: data (approximately) lies on a low-dimensional space of the ambient space

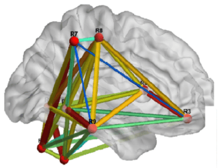
	Ambient space: Euclidean $\mathbb{R}^D$	Ambient space: Manifold M
Less flexible model	Goal: Learn linear subspace… $\mathbb{R}^D = \mathbb{R}^2$:	Goal: Learn "geodesic" submanifold… $M = S^2$: $M = H^2$:
More flexible model	Goal: Learn non-linear subspace… $\mathbb{R}^D = \mathbb{R}^2$:	Goal: Learn "non-geodesic" submanifold… $M = S^2$:  $M = H^2$: 

… and (approximate) posterior distributions for the low-dimensional representation of each data point.

Questions:

From 3D Functional MRI

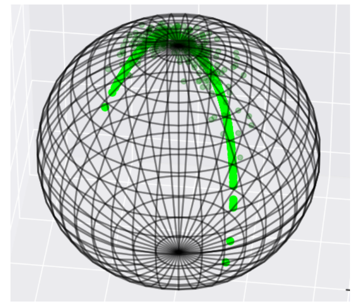




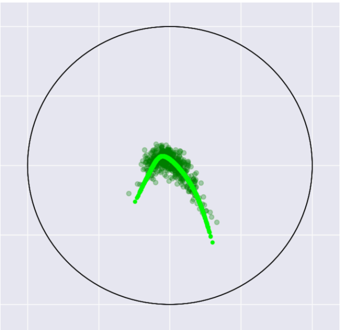
Functional connectome
$\in$ manifold $\mathrm{SPD}(n_{\mathrm{areas}})$
of dimension $n_{\mathrm{areas}}\ (n_{\mathrm{areas}} + 1)$ / 2

Learn "non-geodesic" submanifold…

$M = S^2$:



$M = H^2$:



… and (approximate) posterior distributions for the low-dimensional representation of each data point.

Methodological questions:

- Can we extend traditional dimension reduction methods on Riemannian manifolds to learn "non-geodesic" submanifolds?
- What is the curvature of the learned submanifold: is it flat? [Shao, Kumar, Fletcher 2018].

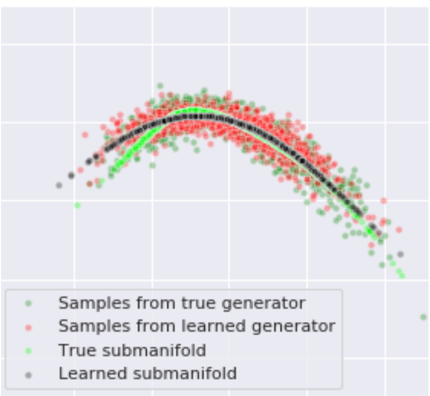
Domain question:

- Do more flexible models provide new insights on brain functional connectomes: is there a pattern in the resting state functional connectomes?

Outline: Learning submanifolds with gVAEs

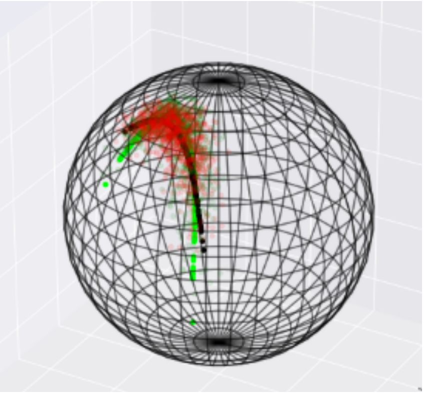
Part 1

Probabilistic PCA, Variational autoencoders and manifold learning



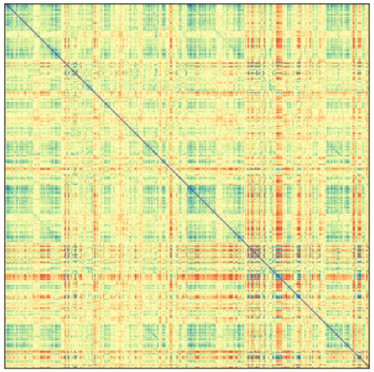
Part 2

Geometric variational autoencoders (gVAEs) and submanifold learning



Part 3

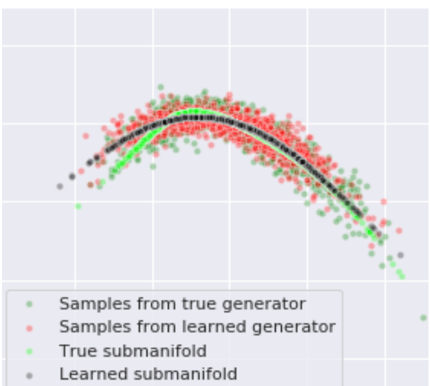
Learning the submanifold of functional brain connectomes



Outline: Learning submanifolds with gVAEs

Part 1

Probabilistic PCA, Variational autoencoders and manifold learning



Part 2

Geometric variational autoencoders (gVAEs) and submanifold learning

Part 3

Learning the submanifold of functional brain connectomes

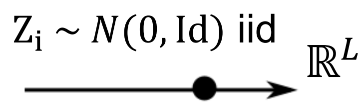
- Probabilistic Principal Component Analysis, EM algorithm.
- Variational Autoencoders, “Amortized Stochastic Variational Gradient EM” algorithm.

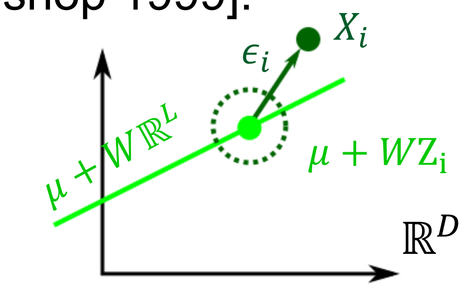
Probabilistic Principal Component Analysis

Generative model of Probabilistic PCA for data in $\mathbb{R}^D$ [Tipping, Bishop 1999]:

$$X_i = \mu + W \mathrm{Z_i} + \epsilon_i$$

- Parameters $\mu \in \mathbb{R}^D, W \in \mathbb{R}^{D \times L}$
- Latent variables $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid



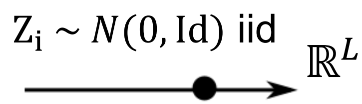


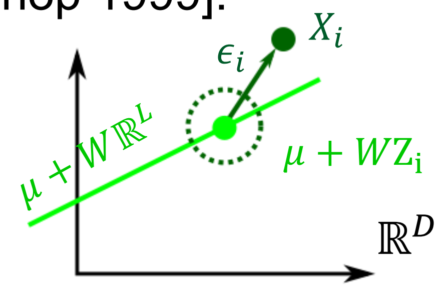
Probabilistic Principal Component Analysis

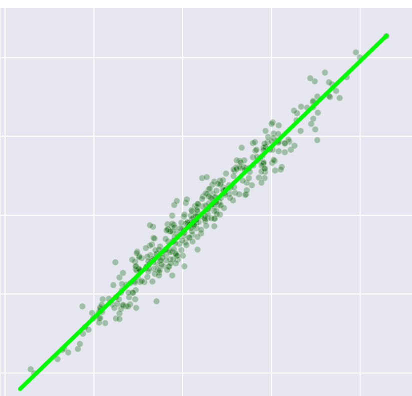
Generative model of Probabilistic PCA for data in $\mathbb{R}^D$ [Tipping, Bishop 1999]:

$$X_i = \mu + W \mathrm{Z_i} + \epsilon_i$$

- Parameters $\mu \in \mathbb{R}^D, W \in \mathbb{R}^{D \times L}$
- Latent variables $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid







Goals of Probabilistic PCA:

- Maximum likelihood (ML) estimation of parameters (W, μ, σ)
  - Learn the linear subspace $\mu + W\mathbb{R}^L$
- Inference on posterior distributions $p_{W,\mu,\sigma}(Z|X_i)$
  - Learn latent variables with uncertainties

Method of Probabilistic PCA:

- Likelihood is not tractable: $p_{W,\mu,\sigma}(x) = \int_Z p_{W,\mu,\sigma}(x, z)dz$.

→ No direct ML estimation of (W, μ, σ) → Expectation-Maximization (EM) algorithm.

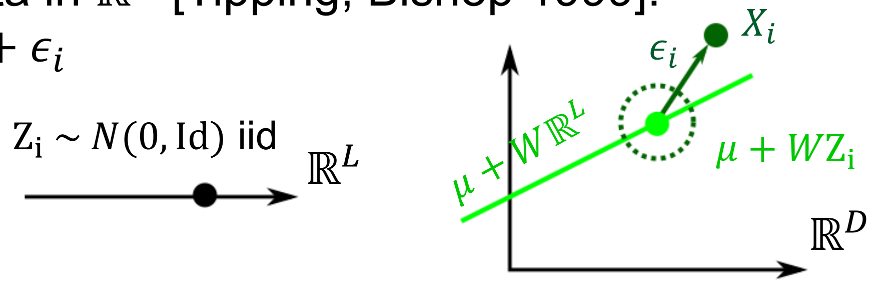
Probabilistic Principal Component Analysis

Generative model of Probabilistic PCA for data in $\mathbb{R}^D$ [Tipping, Bishop 1999]:

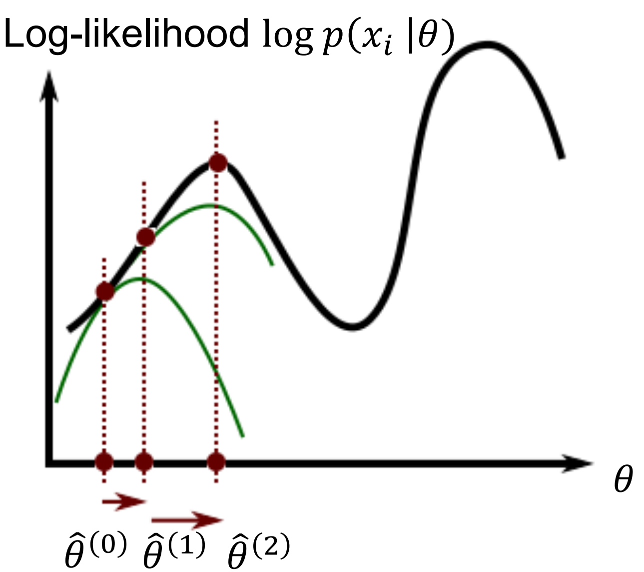
$$X_i = \mu + W Z_i + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid.

Notation: $\theta = (\mu, W, \sigma)$.



EM algorithm for learning and inference in PPCA:



Initialization: : $\hat{\theta}^{(0)}$. Then, iterate until convergence:

1. E-step: At $\hat{\theta}^{(k)}$ fixed, inference on Z:

2. M-step: At $q^{*(k)}(z)$ fixed, maximize lower bound in θ:

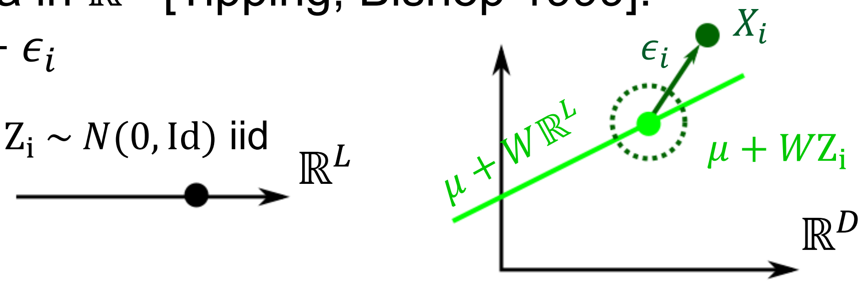
Probabilistic Principal Component Analysis

Generative model of Probabilistic PCA for data in $\mathbb{R}^D$ [Tipping, Bishop 1999]:

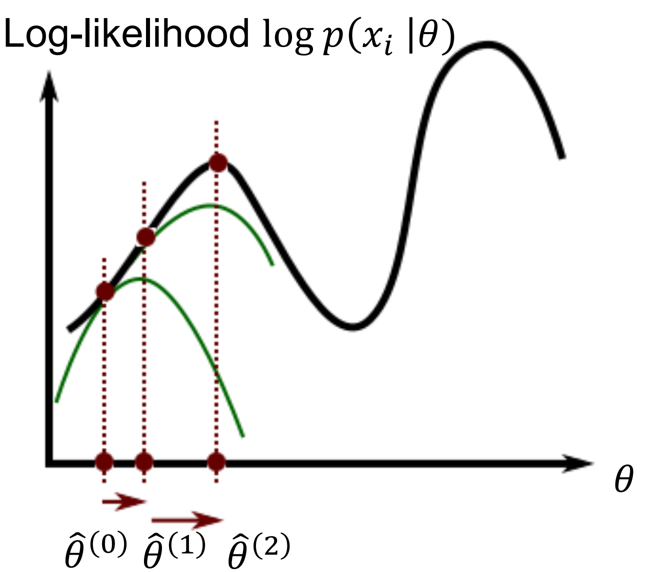
$$X_i = \mu + W Z_i + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid.

Notation: $\theta = (\mu, W, \sigma)$.



EM algorithm for learning and inference in PPCA:



Initialization: : $\hat{\theta}^{(0)}$. Then, iterate until convergence:

1. E-step: At $\hat{\theta}^{(k)}$ fixed, inference on Z:

- $q^{*(k)}(z) = p(z | x_i, \hat{\theta}^{(k)})$ closed form for posterior

2. M-step: At $q^{*(k)}(z)$ fixed, maximize lower bound in θ:

- $\theta \to \int_Z \log \frac{p(x_i, z|\theta)}{q^{*(k)}(z)} . q^*(z)^{(k)} d\mu(z)$

where the inequality:

$$\log p(x_i | \theta) = \log \int_Z p(x_i, z | \theta) d\mu(z) \geq \int_Z \log \frac{p(x_i, z|\theta)}{q(z)} . q(z) d\mu(z)$$

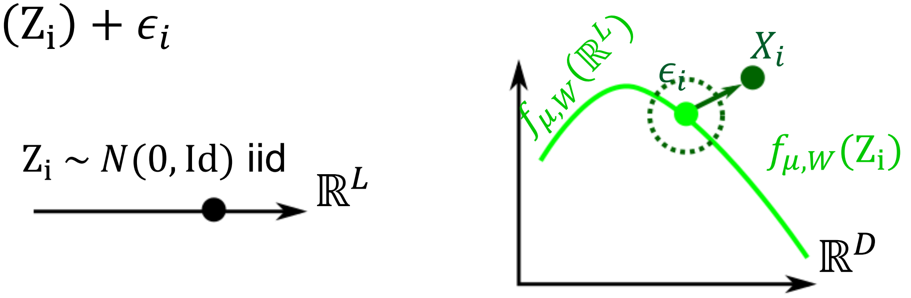
is valid for any q and is tangent at $\hat{\theta}^{(k)}$ for $q^{*(k)}(z)$.

Variational Autoencoders (VAEs)

Generative model of Variational Autoencoders for data in $\mathbb{R}^D$ [Kingma, Welling 2014]:

$$X_i = f_{\mu,W}(\mathrm{Z_i}) + \epsilon_i$$

- Parameters: $\boldsymbol{\mu}, \boldsymbol{W}$
- Latent variables: $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid



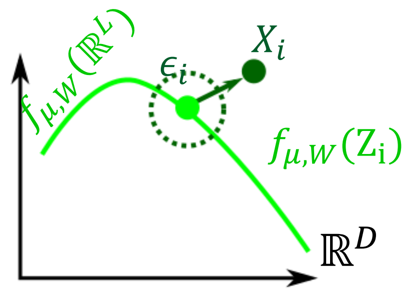
- Function: $f_{\mu,W}(\mathrm{Z_i}) = \Pi_{k=1}^{K} \sigma_k(W_k \cdot \ + \mu_k)$ fully connected neural network, K layers.

Variational Autoencoders (VAEs)

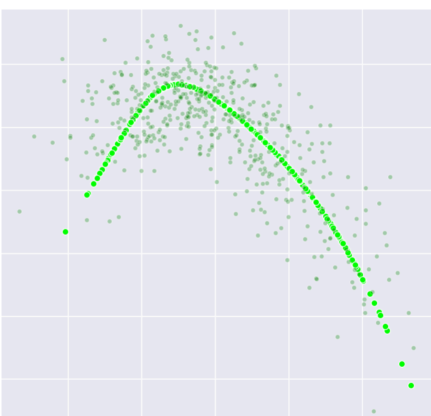
Generative model of Variational Autoencoders for data in $\mathbb{R}^D$ [Kingma, Welling 2014]:

$$X_i = f_{\mu,W}(\mathrm{Z_i}) + \epsilon_i$$

- Parameters: $\boldsymbol{\mu}, \boldsymbol{W}$
- Latent variables: $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid



- Function: $f_{\mu,W}(\mathrm{Z_i}) = \Pi_{k=1}^{K} \sigma_k(W_k \cdot\ + \mu_k)$ fully connected neural network, K layers.



Goals of Variational Autoencoders:

- Maximum likelihood (ML) estimation of parameters θ
  - Learn the **non-linear** subspace $f_{\boldsymbol{\mu},\boldsymbol{W}}(\mathbb{R}^L)$
- Inference on posterior distributions $p_{\boldsymbol{\mu},\boldsymbol{W}}(Z|X_i)$
  - Learn latent variables with uncertainties

Method of Variational Autoencoders:
→ Expectation-Maximization (EM) algorithm?

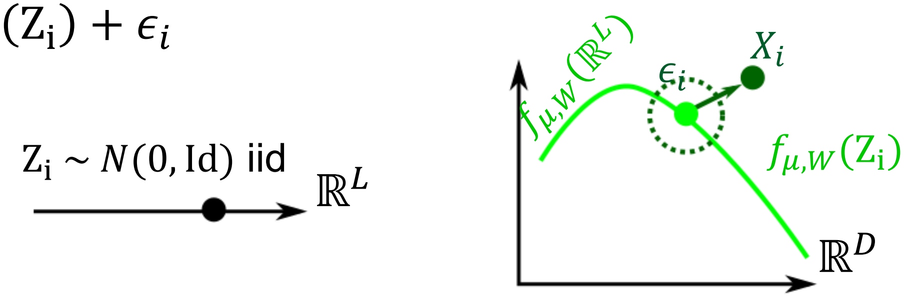
Variational Autoencoders (VAEs)

Generative model of Variational Autoencoders for data in $\mathbb{R}^D$ [Kingma, Welling 2014]:

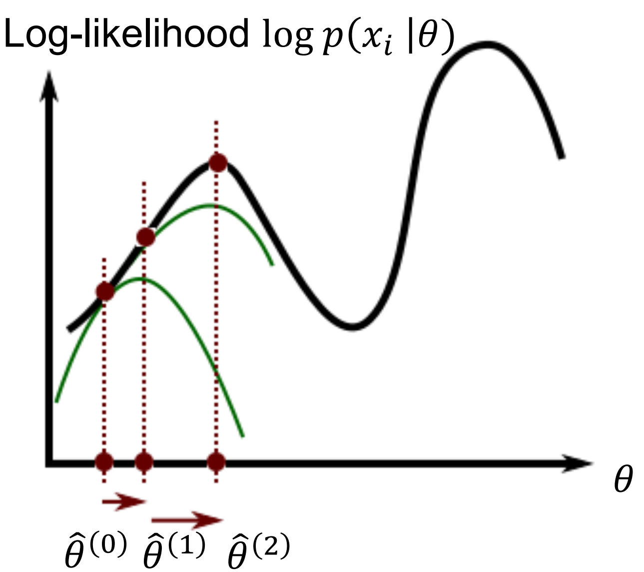
$$X_i = f_{\mu,W}(\mathrm{Z_i}) + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid

Notation: $\theta = (\mu, W, \Psi)$.



EM algorithm?



Initialization: : $\hat{\theta}^{(0)}$. Then, iterate until convergence:

1. E-step: At $\hat{\theta}^{(k)}$ fixed, inference on Z?

2. M-step: At $q^{*(k)}(z)$ fixed, maximize lower bound in θ?

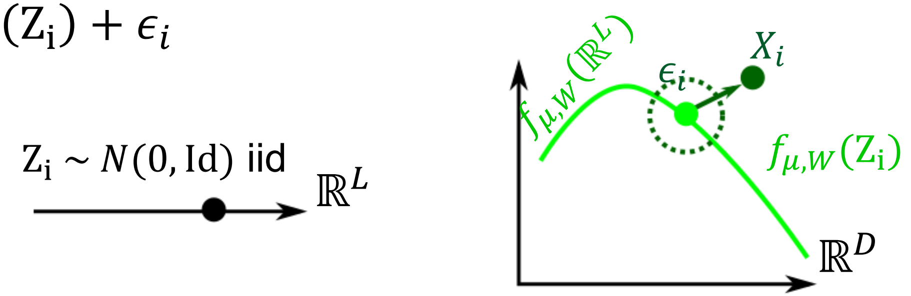
Variational Autoencoders (VAEs)

Generative model of Variational Autoencoders for data in $\mathbb{R}^D$ [Kingma, Welling 2014]:

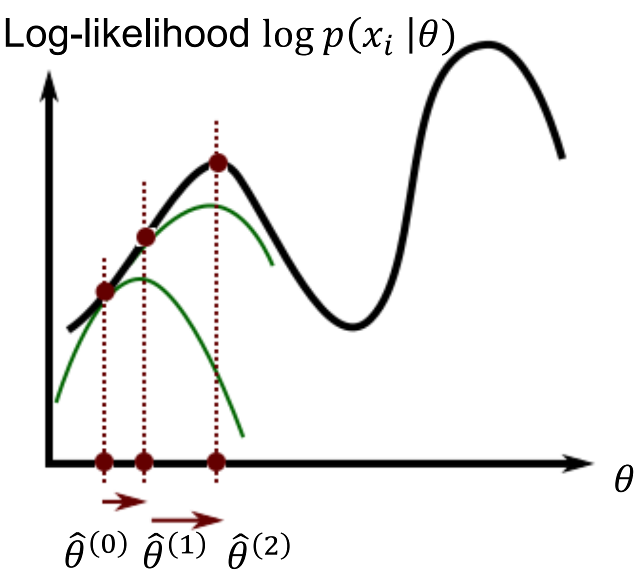
$$X_i = f_{\mu,W}(\mathrm{Z_i}) + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid

Notation: $\theta = (\mu, W, \Psi)$.



EM algorithm?



Initialization: : $\hat{\theta}^{(0)}$. Then, iterate until convergence:

1. E-step: At $\hat{\theta}^{(k)}$ fixed, inference on Z?

→ **Problem:** The posterior does not have a closed form.

→ We cannot compute the tangent lower bound.

2. M-step: At $q^{*(k)}(z)$ fixed, maximize lower bound in θ?

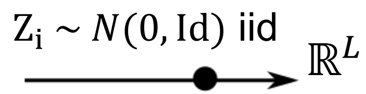
Variational Autoencoders (VAEs)

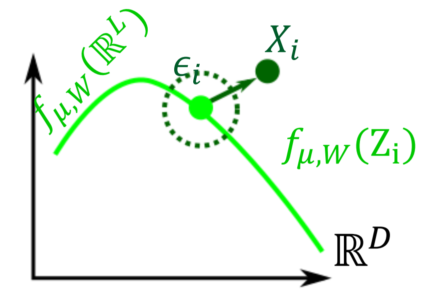
Generative model of Variational Autoencoders for data in $\mathbb{R}^D$ [Kingma, Welling 2014]:

$$X_i = f_{\mu,W}(\mathrm{Z_i}) + \epsilon_i$$

with $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid

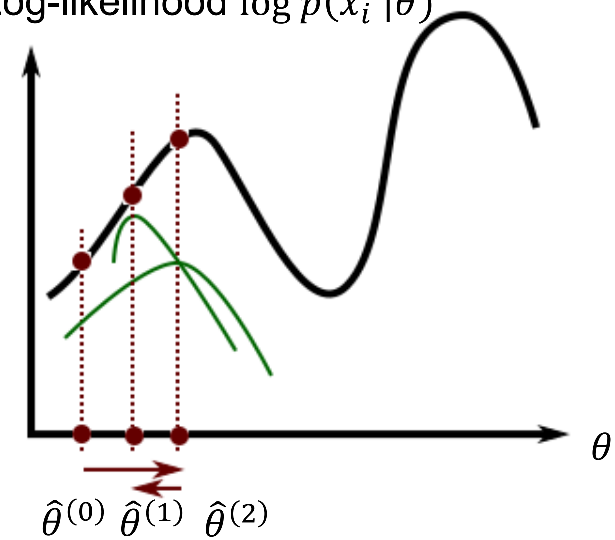
Notation: $\theta = (\mu, W, \Psi)$.





Variational EM algorithm?

Log-likelihood $\log p(x_i \mid \theta)$



Initialization: : $\hat{\theta}^{(0)}$. Then, iterate until convergence:

1. Variational E-step: At $\hat{\theta}^{(k)}$ fixed, inference on Z?
Variational Inference: Find the distribution $q^{*(k)}(z)$ within a tractable variational family, that is the closest to the posterior.

2. M-step: At $q^{*(k)}(z)$ fixed, maximize lower bound in θ?
Only an approximation of the posterior:
→ the lower bound at $\hat{\theta}^{(k)}$ is not tangent.

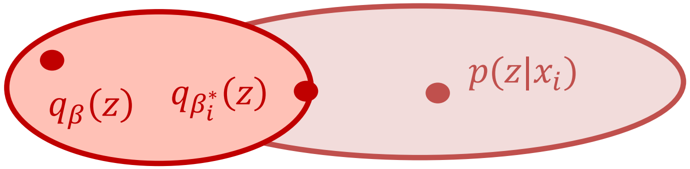
Variational EM

- **E-step:** Untractable posterior → HMC approximation or Variational Inference (here)

Variational Inference:

- Choose a family of densities: $Q = \{q_\beta \,|\beta \in B\}$.
- Find $q_i^* = q_{\beta_i^*} \in Q$ as close as possible to $p(z|x_i)$ where "close" is by Kullback-Leibler DV.

$$\beta_i^* = \text{argmin}_{\beta \in B} KL\big(q_\beta(z) \;||\; p(z|x_i)\big)$$
$$= \text{argmax}_{\beta \in B} \log p_{\widehat{\theta_k}}(x_i) - KL\big(q_\beta(z) \,||\, p(z|x_i)\big)$$
$$= \text{argmax}_{\beta \in B} \text{ELBO}\big(\widehat{\theta_k}, \beta, x_i\big)$$



- **M-step:** Only an approximation of the posterior, thus the lower bound at $\hat{\theta}^{(k)}$ is not tangent.

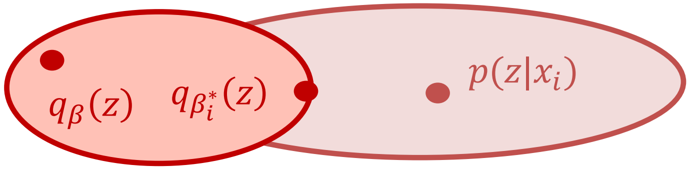
Variational EM

- **E-step:** Untractable posterior → HMC approximation or Variational Inference (here)

Variational Inference:

- Choose a family of densities: $\mathrm{Q} = \{q_\beta \mid \beta \in \mathrm{B}\}$.
- Find $q_i^* = q_{\beta_i^*} \in Q$ as close as possible to $p(z|x_i)$ where "close" is by Kullback-Leibler DV.

$$\begin{aligned}\beta_i^* &= \operatorname{argmin}_{\beta\in\mathrm{B}} KL\big(q_\beta(z) \;||\; p(z|x_i)\big)\\ &= \operatorname{argmax}_{\beta\in\mathrm{B}} \log p_{\widehat{\theta_k}}(x_i) - KL\big(q_\beta(z) \,||\, p(z|x_i)\big)\\ &= \operatorname{argmax}_{\beta\in\mathrm{B}} \mathrm{ELBO}\big(\widehat{\theta_k}, \beta, x_i\big)\end{aligned}$$



- **M-step:** Only an approximation of the posterior, thus the lower bound at $\hat{\theta}^{(k)}$ is not tangent.

$$L(\theta) = \log p_\theta(x_i) \geq \int_z \log \frac{p(x_i, z|\theta)}{q_{\beta_i^*}(z)} \cdot q_{\beta_i^*}(z) d\mu(z) = \underbrace{\log p_\theta(x_i) - \mathrm{KL}\big(q_{\beta_i^*}(z) \,\|\, p(z|x_i)\big)}_{\mathrm{ELBO}(\theta, \beta_i^*, x_i)}$$

$$\hat{\theta} = \operatorname{argmax}_\theta \mathrm{ELBO}(\theta, \{\beta_i^*\}_i, \{x_i\}_i)$$

→ At each iteration, (n+1) maximizations of the same criterion: $\mathrm{ELBO}(\theta, \{\beta_i\}_i, \{x_i\}_i)$
→ VAE algorithm: at each iteration, 2 gradients steps of the same criterion.

Variational Autoencoders

= "Stochastic Amortized Variational Gradient EM"

Fix a parametric family: $Q = \{q_\beta ; \beta \in \mathrm{B}\}$, and iterate two steps:

- ``Amortized gradient E-step". $\theta^{(k)}$ fixed.

Learn a function $g_\phi : x_i \to g_\phi(x_i)$ that predicts the optimal parameter of the variational inference: $g_\phi(x_i) = \widehat{\beta_i^*}$ that estimates $\beta_i^* = \mathrm{argmin}_{\beta \in \mathrm{B}} KL\left(q_\beta(z) \mid\mid p(z|x_i)\right)$

$$\phi^{(k+1)} = \phi^{(k)} - \eta \nabla_\phi KL(q_{g_\phi(x_i)}(z) \mid\mid p_{\theta^{(k)}}(z_i|x_i))$$
$$= \phi^{(k)} + \eta \nabla_\phi \mathrm{ELBO}(\theta^{(k)}, \phi, x_i)$$

- Gradient M-step: $\phi^{(k+1)}$ fixed.

$$\theta^{(k+1)} = \theta^{(k)} + \eta \nabla_\theta \mathrm{ELBO}\left(\theta, \phi^{(k+1)}, x_i\right)$$

Stochastic gradient ascent in (ϕ, θ) on ELBO

Where: $\mathrm{ELBO}(\theta, \phi, x_i) = \log p_\theta(x_i) - \mathrm{KL}\left(q_{g_{\phi(x_i)}}(z) \parallel p(z|x_i)\right)$

And $\mathrm{ELBO}(\theta, \phi, x_i)$ can be conveniently rewritten as:

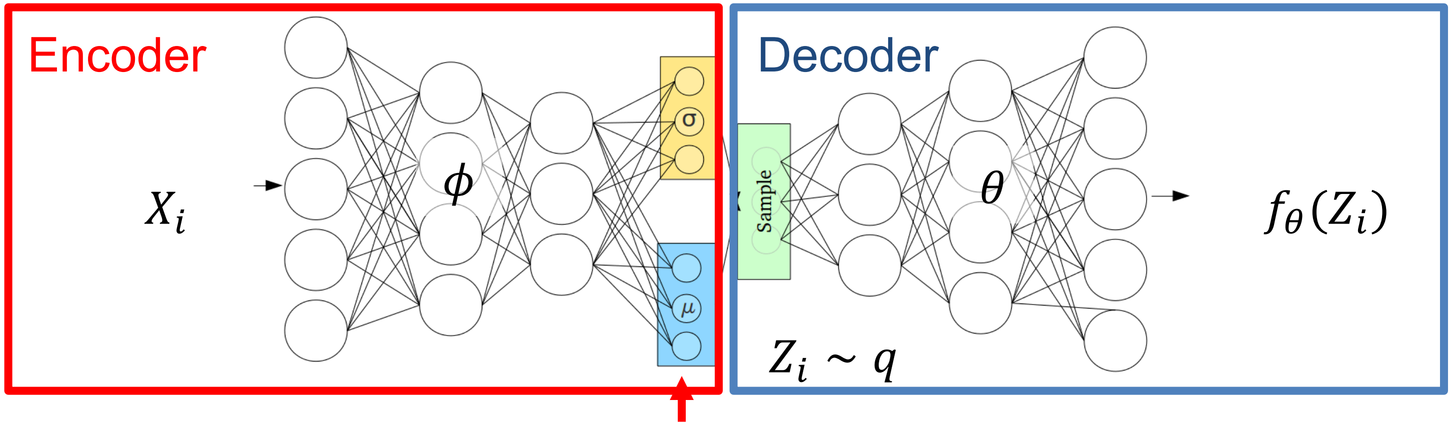
$$\mathrm{ELBO}(\theta, \phi, x_i) = \mathbb{E}_{q_{g_{\phi(x_i)}}}\left(\log p_\theta(x_i|z)\right) - \mathrm{KL}\left(q_{g_{\phi(x_i)}}(z) \parallel p(z)\right)$$

tractable via variational family

Given by the generative model

Parameterization with two NNs

We can model g_ϕ and f_θ as neural networks with parameters ϕ and θ.



$g_\phi(X_i)$ that parameterizes $q_{g_\phi(X_i)}$ in multidimensional diagonal Gaussian

Train them simultaneously on:

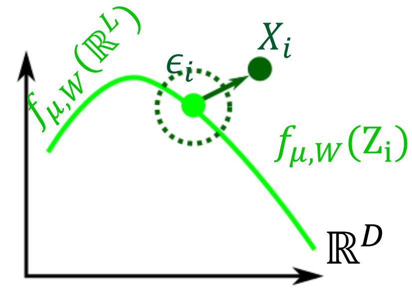
$$\mathrm{ELBO}(\theta, \phi, x) = \mathbb{E}_{q_{g_{\phi(x)}}}\left(\log p_\theta(x|z)\right) - \mathrm{KL}\left(q_{g_{\phi(x)}}(z) \parallel p(z)\right)$$

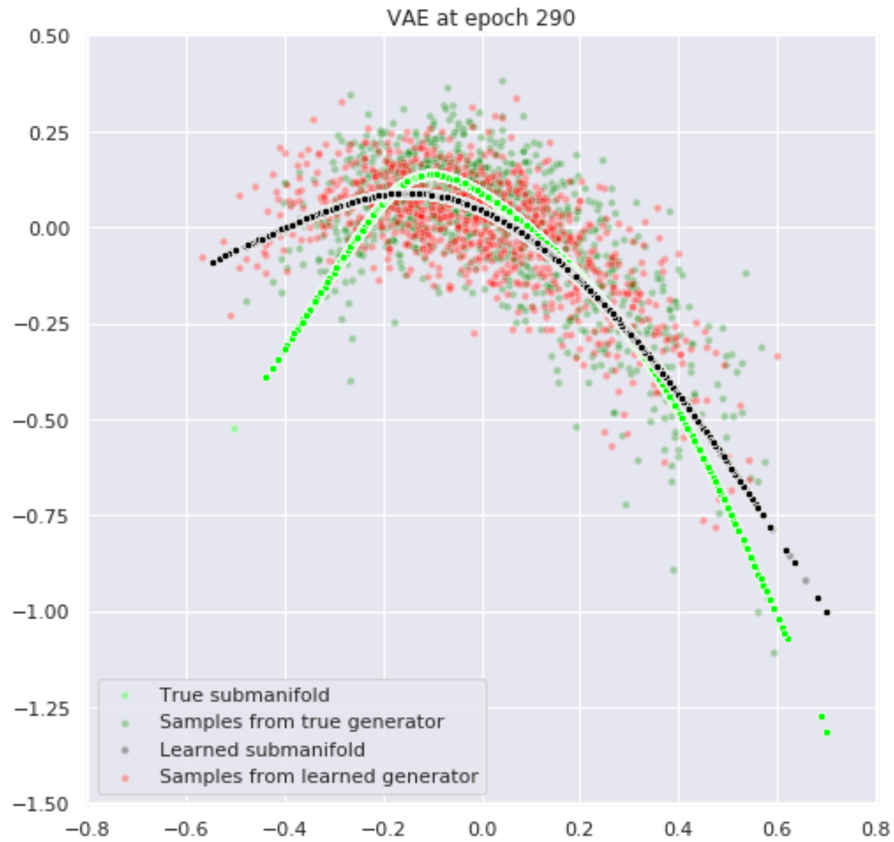
tractable via variational family

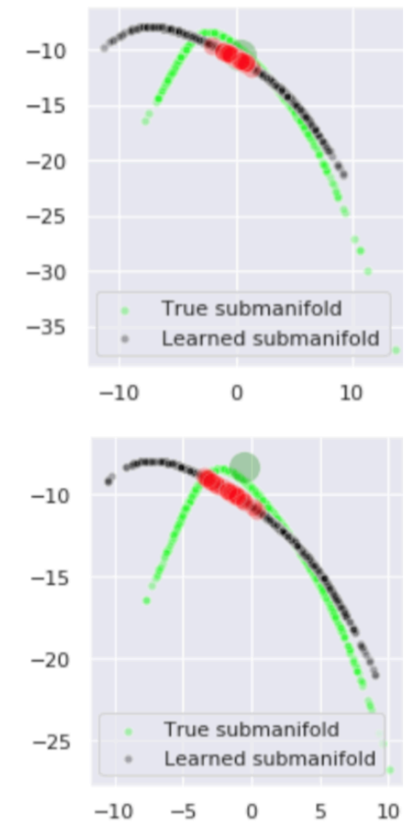
Given by the generative model

Training VAE

$Z_i \sim N(0, \mathrm{Id})$ iid $\mathbb{R}^L$







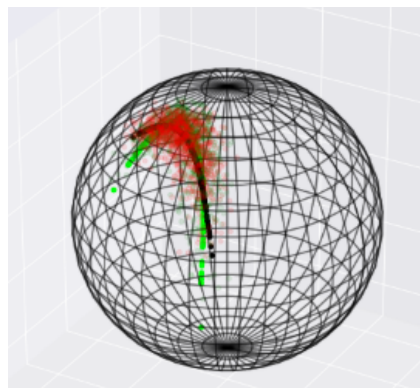
Outline: Learning submanifolds with gVAEs

Part 1

Probabilistic PCA, Variational autoencoders and manifold learning

Part 2

Geometric variational autoencoders (gVAEs) and submanifold learning



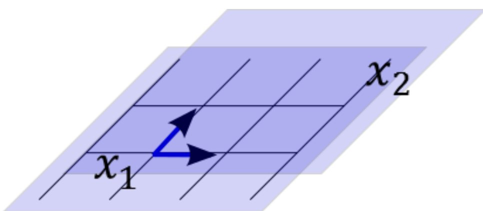
Part 3

Learning the submanifold of functional brain connectomes

- Elements of Geometric Statistics
- Geometric VAEs and geometry of learned submanifold (“latent space”)
- Is the learned submanifold flat? [Shao, Kumar, Fletcher 2018]

Elements of Geometric Statistics

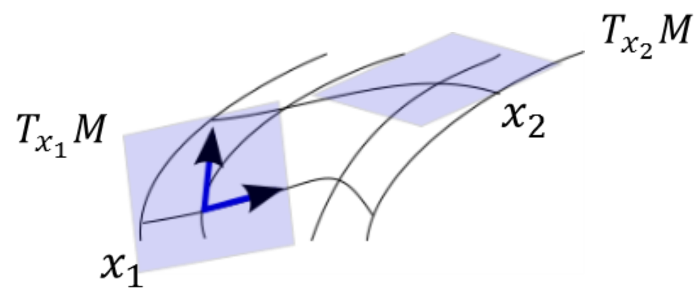
Vector space $\mathbb{R}^D$



- Euclidean space:

Add (global) inner product

Manifold M

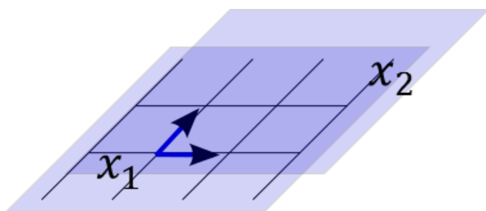


- Riemannian manifold:

Add local inner products = a Riemannian metric

Elements of Geometric Statistics

Vector space $\mathbb{R}^D$

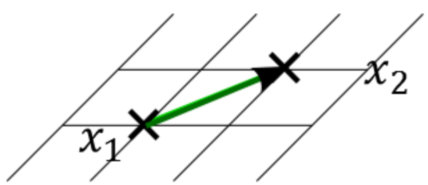


- Euclidean space:

Add (global) inner product

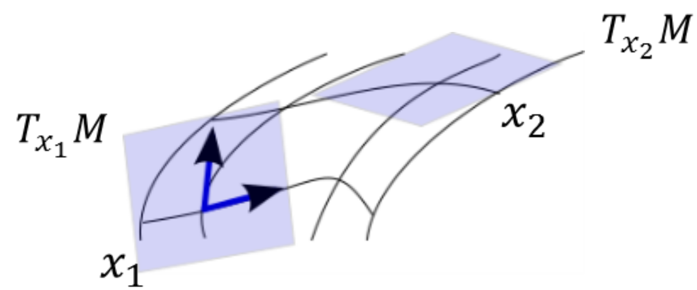
- Straight-line: minimal-length curve

$\mathrm{dist}(x_1, x_2)$: length of the line



Adding vector $\overrightarrow{x_1 x_2}$ to x_1 : $\mathrm{x}_2 = \mathrm{x}_1 + \overrightarrow{x_1\, x_2}$
Subtracting x_2 to x_1 : $\overrightarrow{x_1 x_2} = \mathrm{x}_2 - \mathrm{x}_1$

Manifold M

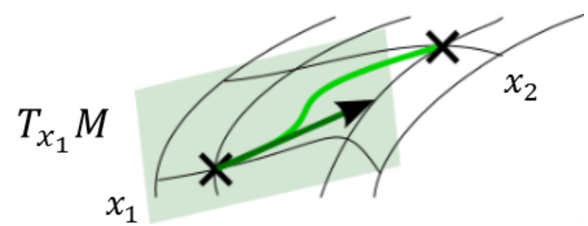


- Riemannian manifold:

Add local inner products = a Riemannian metric

- Riemannian geodesic: minimal length curve

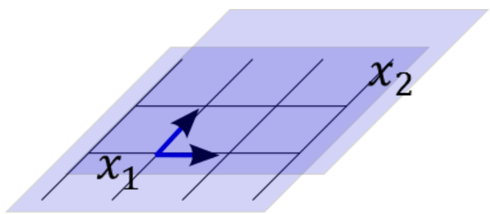
$\mathrm{dist}_{\mathrm{M}}(x_1, x_2)$: length of the geodesic

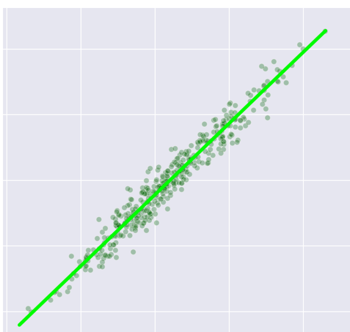


Exponentiating vector $\overrightarrow{x_1 x_2}$ from x_1: $\mathrm{x}_2 = Exp_{x_1}(\overrightarrow{x_1 x_2})$
Taking the logarithm of x_2 at x_1: $\overrightarrow{x_1 x_2} = Log_{x_1}(x_2)$

Elements of Geometric Statistics

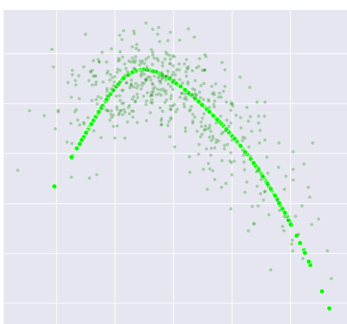
Vector space $\mathbb{R}^D$





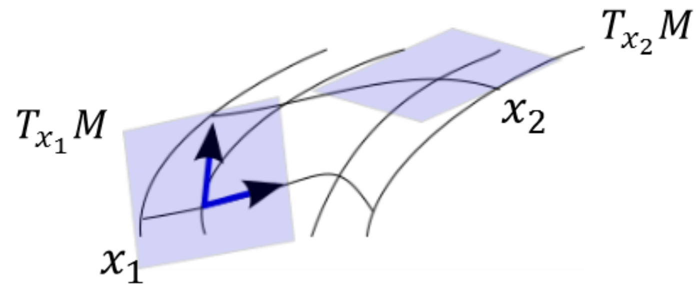
Linear subspace:

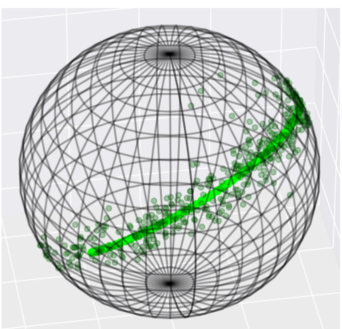
- point
- basis at that point



Non-linear subspace

Manifold M

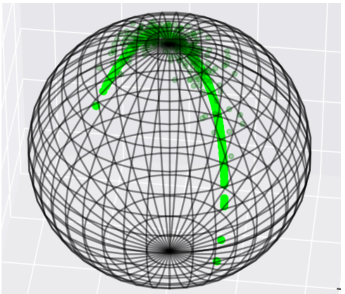




"Geodesic submanifold":

- point
- basis at that point

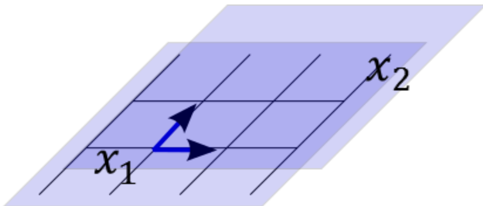
= Submanifold that is geodesic at a point



Non-geodesic submanifold

Elements of Geometric Statistics

Vector space $\mathbb{R}^D$

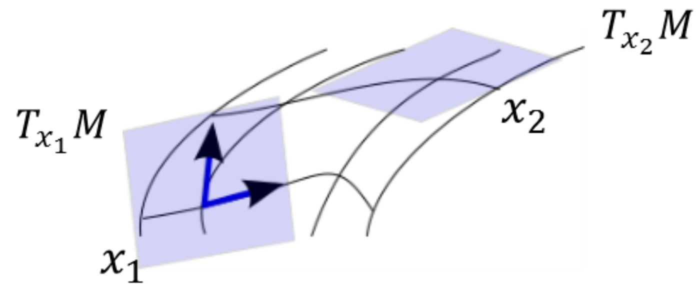


- Volume measure: dx

- Isotropic normal distribution:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{||x - \mu||^2}{2\sigma^2}\right)$$

Manifold M



- Volume measure at x: $dM(x) = \det\sqrt{Z(x)}dx$ where Z is the matrix defining the inner product.

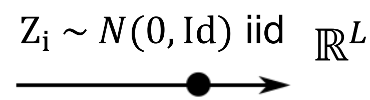
- Riemannian isotropic normal distribution:

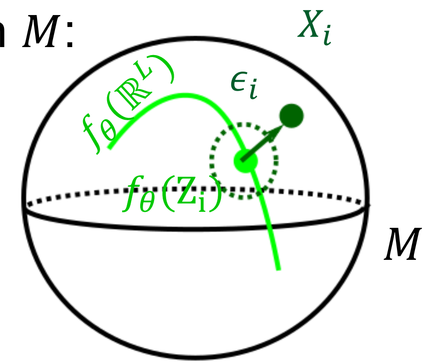
$$p(x) = C_M(\mu, \sigma) \exp\left(-\frac{\text{dist}_M^2(x, \mu)}{2\sigma^2}\right)$$

Geometric Variational Autoencoders (gVAEs)

Generative model of geometric Variational Autoencoders for data in M:

$$X_i = \mathrm{Exp}^{\mathrm{M}}(f_\theta(\mathrm{Z_i}), \epsilon_i)$$



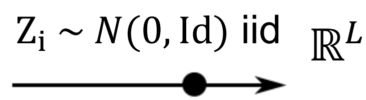


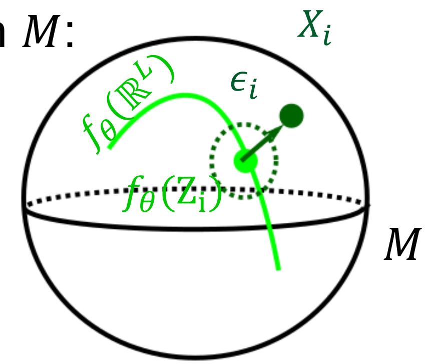
- Parameters: $\boldsymbol{\mu}, \boldsymbol{W}$
- Latent variables: $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise: $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid
- Function: $f_{\mu,W}(\mathrm{Z_i}) = \Pi_{k=1}^{K} \sigma_k(W_k \cdot + \mu_k)$ fully connected neural network, K layers.

Geometric Variational Autoencoders (gVAEs)

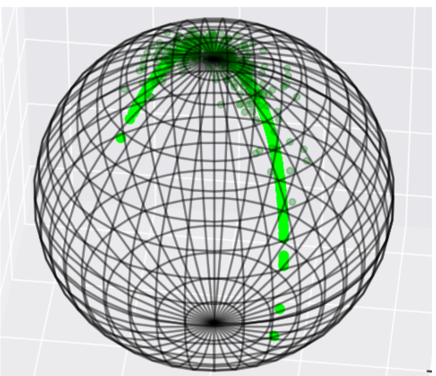
Generative model of geometric Variational Autoencoders for data in M:

$$X_i = \mathrm{Exp}^{\mathrm{M}}(f_\theta(\mathrm{Z_i}), \epsilon_i)$$





- Parameters: $\boldsymbol{\mu}, \boldsymbol{W}$
- Latent variables: $\mathrm{Z_i} \sim N(0, \mathrm{Id})$ iid
- Noise: $\epsilon_i \sim N(0, \sigma^2 \mathrm{Id}_D)$ iid
- Function: $f_{\mu,W}(\mathrm{Z_i}) = \Pi_{k=1}^{K} \sigma_k(W_k \cdot + \mu_k)$ fully connected neural network, K layers.



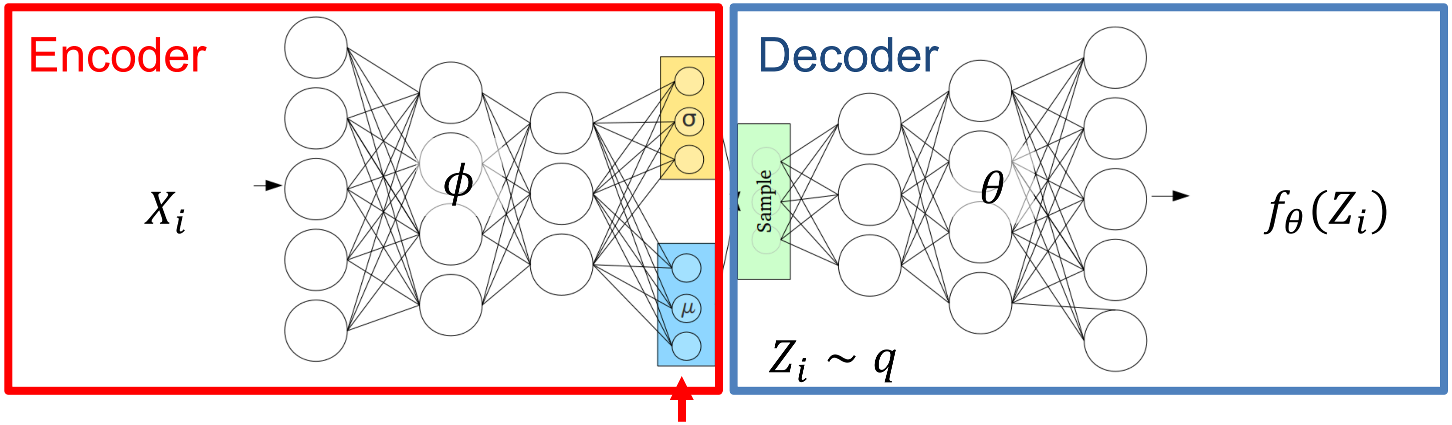
Goals of geometric Variational Autoencoders:

- Maximum likelihood (ML) estimation of parameters θ
  - Learn the **non-geodesic** subspace $f_{\boldsymbol{\mu},\boldsymbol{W}}(\mathbb{R}^L)$
- Inference on posterior distributions $p_{\boldsymbol{\mu},\boldsymbol{W}}(Z|X_i)$
  - Learn latent variables with uncertainties

Method of geometric Variational Autoencoders:
→ Adapt learning from Variational Autoencoders.

Parameterization with two NNs

We can model g_ϕ and f_θ as neural networks with parameters ϕ and θ.



$g_\phi(X_i)$ that parameterizes $q_{g_\phi(X_i)}$ in multidimensional diagonal Gaussian

Train them simultaneously on:

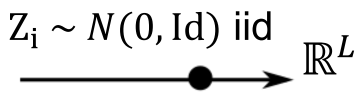
$$\mathrm{ELBO}(\theta, \phi, x) = \mathbb{E}_{q_{g_{\phi(x)}}}\left(\log p_\theta(x|z)\right) - \mathrm{KL}\left(q_{g_{\phi(x)}}(z) \parallel p(z)\right)$$

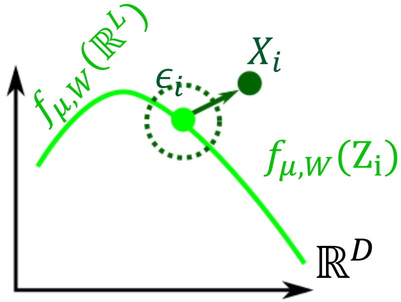
tractable via variational family

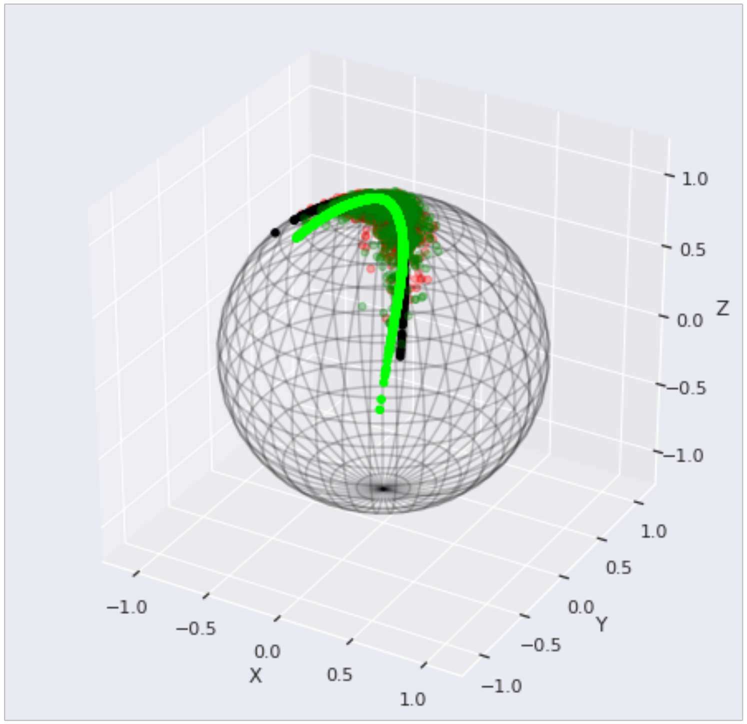
Given by the generative model

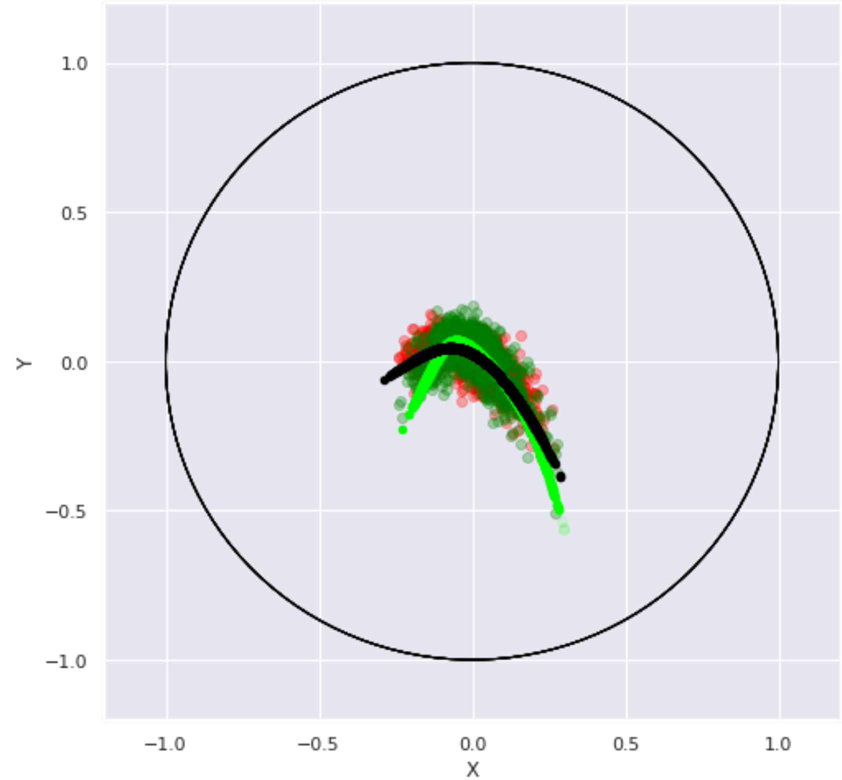
→ $p_\theta(x|z)$ is a Riemannian normal: $p_\theta(x|z) = C_M(\mu, \sigma) \exp\left(-\frac{\mathrm{dist}_M^2(x, f_\theta(z))}{2\sigma^2}\right)$

Training gVAE

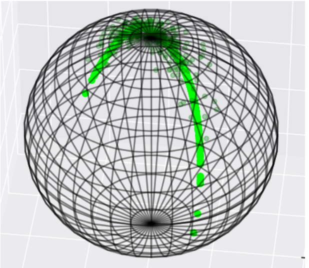








Geometry of the learned submanifold



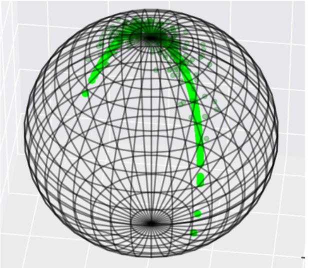
Estimate of the submanifold: $\hat{N} = f_{\hat{\mu},\hat{W}}(\mathbb{R}^L)$.

- **Dimension of $\hat{N}$?**

- **Geometry of $\hat{N}$?** [Kuhnel, Fletcher, Joshi, Sommer 2018]

- **No curvature?** [Shao, Kumar, Fletcher 2018].

Geometry of the learned submanifold



Estimate of the submanifold: $\hat{N} = f_{\hat{\mu},\hat{W}}(\mathbb{R}^L)$.

- **Dimension of $\hat{N}$?**

If the differential $df_{\hat{\mu},\hat{W}}$ is of full rank:

$\dim \hat{N} = L'$, dimension of the space spanned by the latent variables within $\mathbb{R}^L$.

- **Geometry of $\hat{N}$?** [Kuhnel, Fletcher, Joshi, Sommer 2018]

$\hat{N}$ inherits differential geometric structure from the ambient manifold, by pull-back of the Riemannian metric of M:

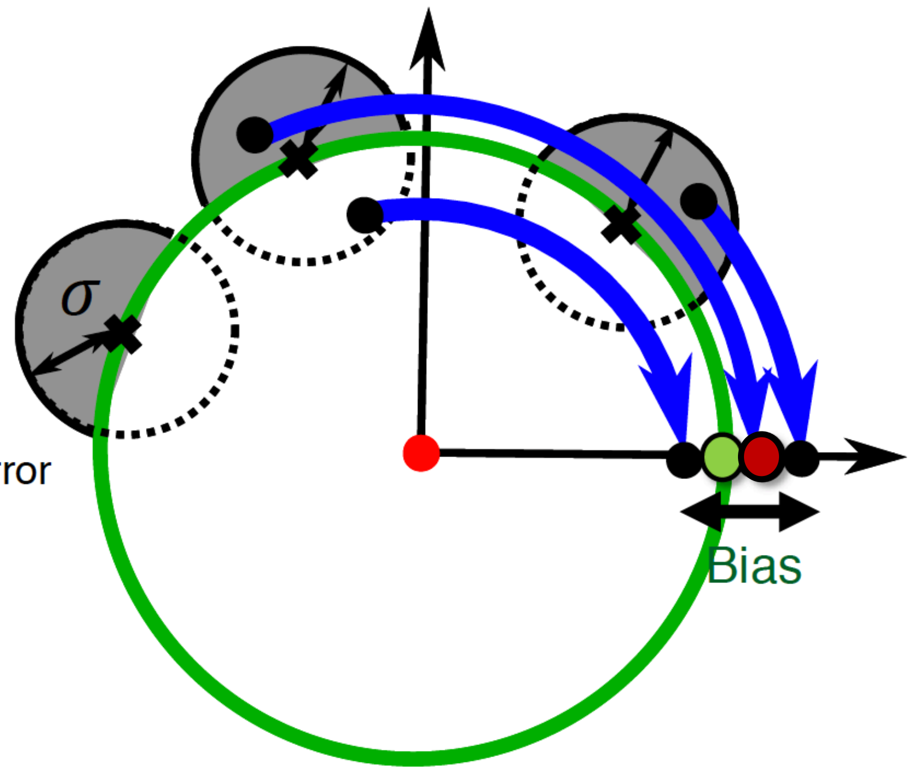
$$\text{for } v, w \in T\_x(\hat{N}) \text{ , } < v, w >_{T_x \hat{N}} = < v, w >_{T_x M}$$

→ In particular, its curvature can be computed.

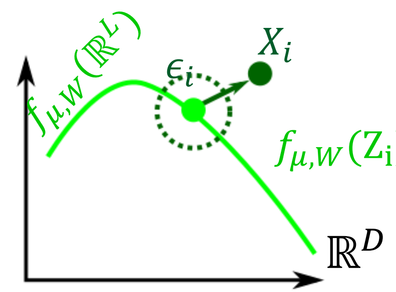
- **No curvature?** [Shao, Kumar, Fletcher 2018].

"Our experiments show that these models represent real image data with manifolds that have surprisingly little curvature.[…] Further investigation into this phenomenon is warranted."

Estimating the Geometry of an Equivalence Class?



σ^2: variance of measurement error



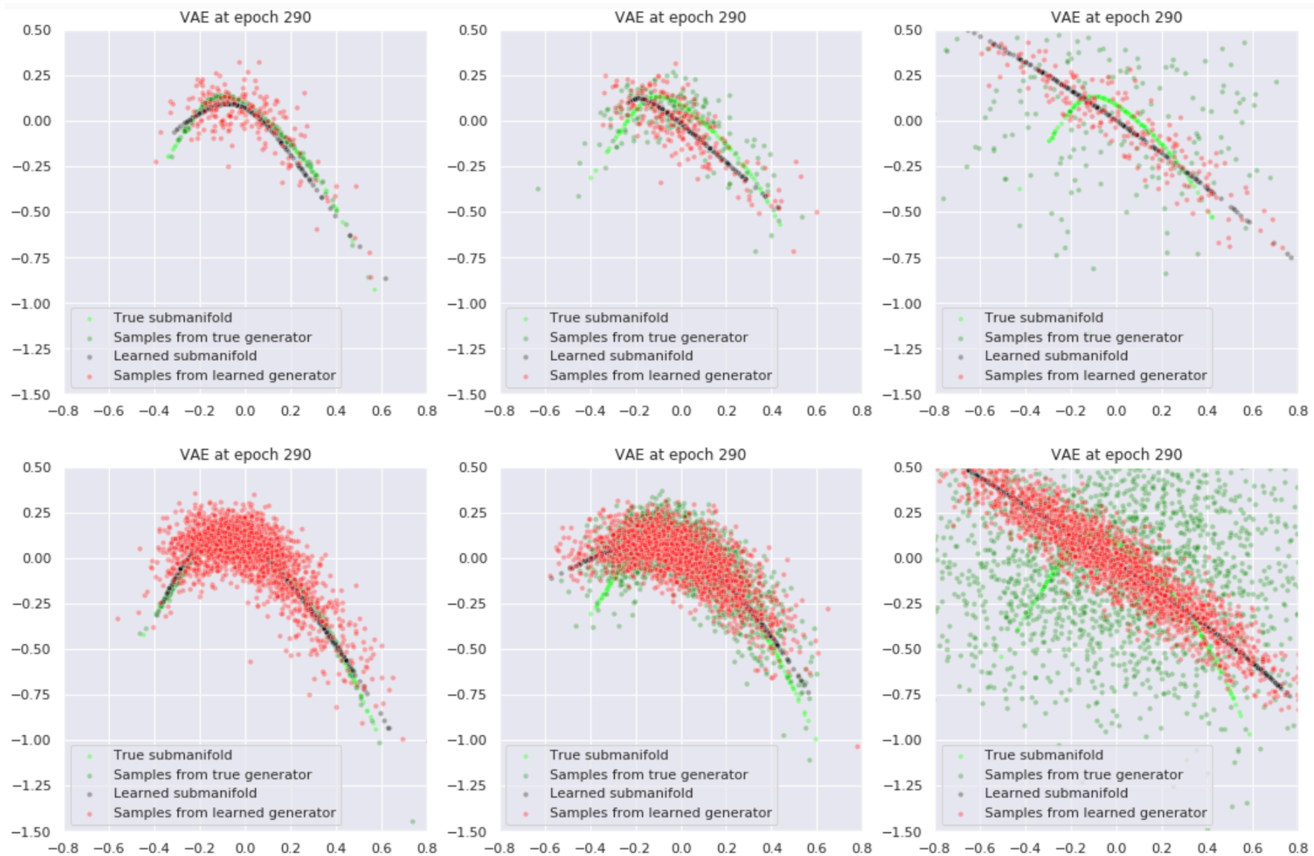
The curvature of submanifold defining the **equivalence class**, at the scale of σ, creates an asymptotic bias [Miolane, Holmes, Pennec 2017]. → **Same behavior for gVAE?**

Estimating the Geometry

$\log \sigma^2 = -10$ | $\log \sigma^2 = -5$ | $\log \sigma^2 = -2$

$n = 10k$

$n = 100k$



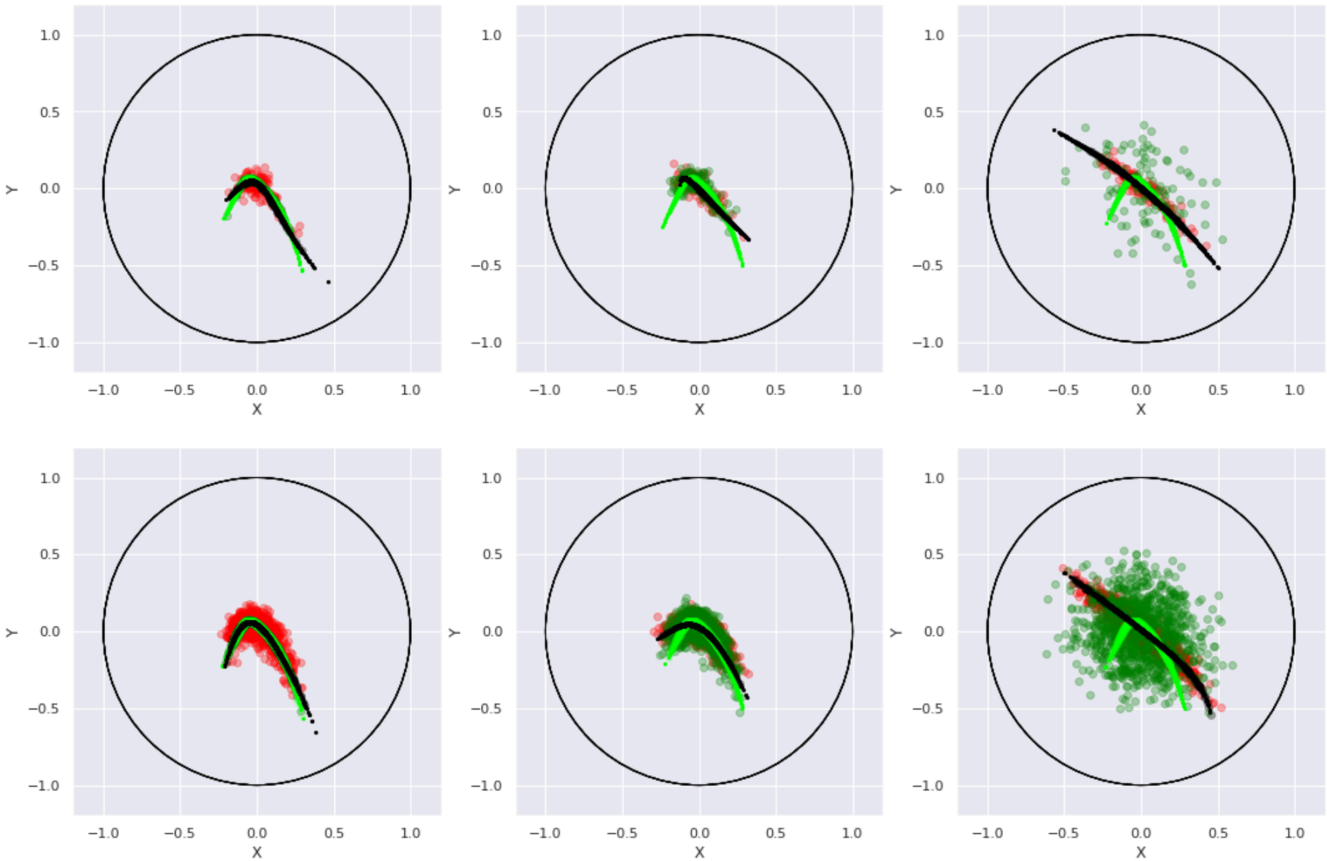
Asymptotic bias of the geometry's estimate,
controlled by the standard deviation of the noise.

Estimating the Geometry

	$\log \sigma^2 = -10$	$\log \sigma^2 = -5$	$\log \sigma^2 = -2$

$n = 10k$

$n = 100k$



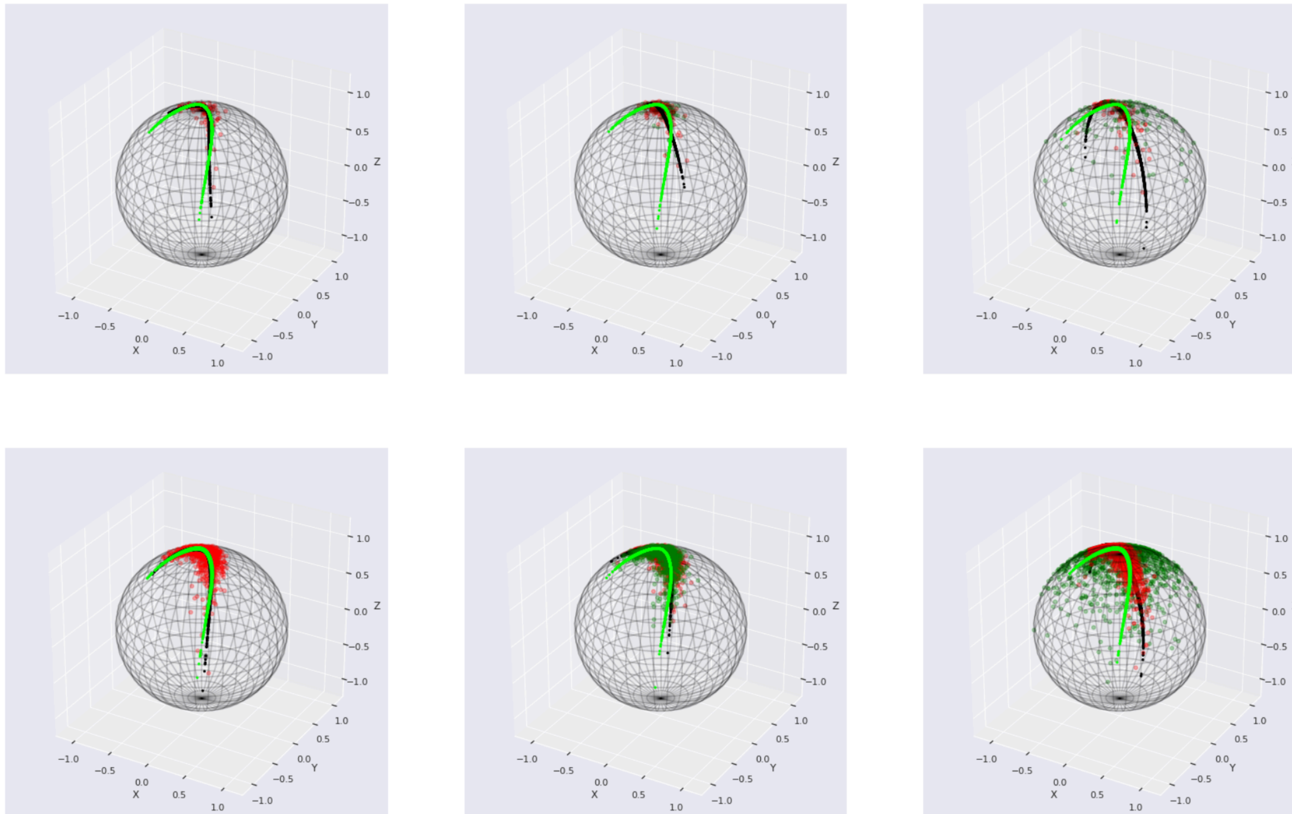
Asymptotic bias of the geometry's estimate, controlled by the standard deviation of the noise.

Estimating the Geometry

$\log \sigma^2 = -10$ | $\log \sigma^2 = -5$ | $\log \sigma^2 = -2$

$n = 10k$

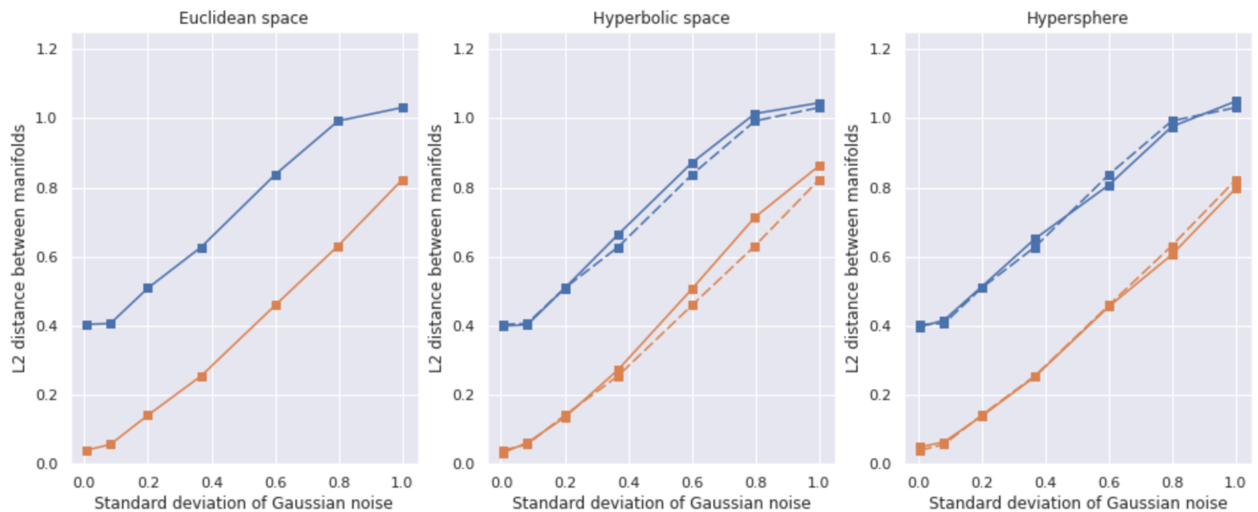
$n = 100k$



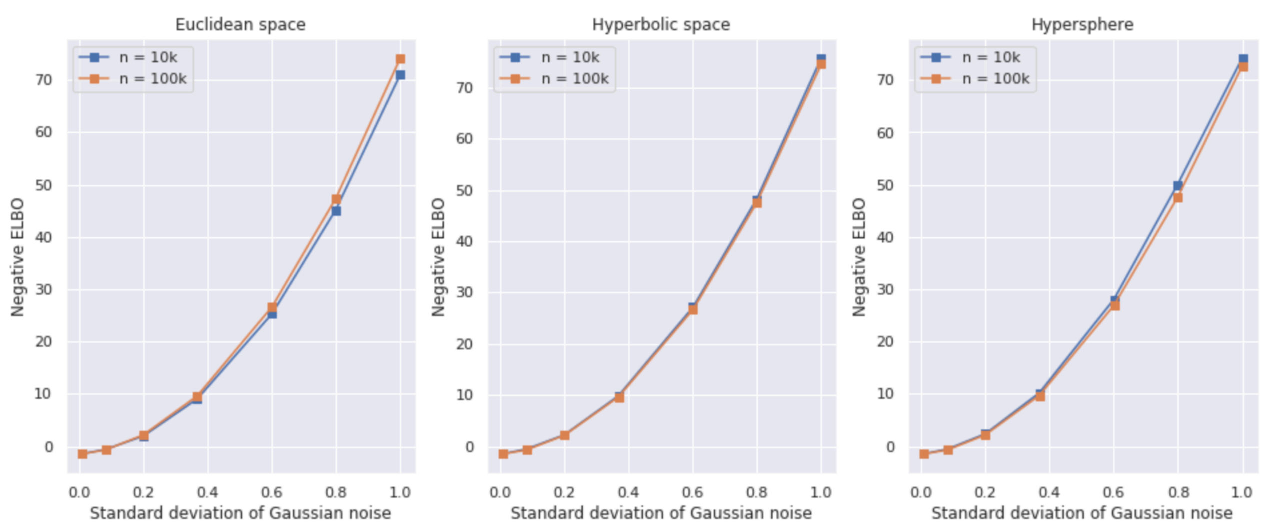
Asymptotic bias of the geometry's estimate,
controlled by the standard deviation of the noise.

Estimating the Geometry

$$\mathrm{dist}(N, \widehat{N}) = \int_z \mathrm{dist}_M\left(f_\theta(z), f_{\widehat{\theta}}(z)\right) p(z) dz$$



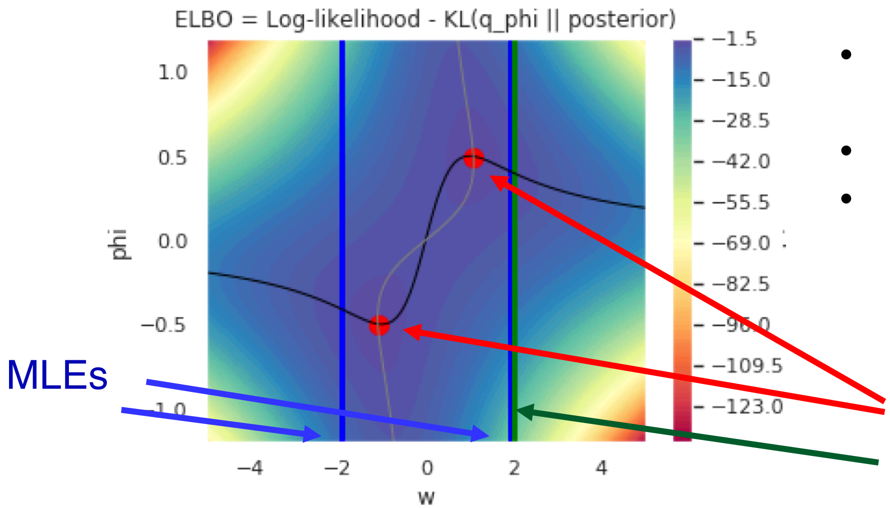
$$\mathrm{ELBO}(\theta, \phi, x) = \mathbb{E}_{q_{g_{\phi(x)}}}\left(\log p_\theta(x|z)\right) - \mathrm{KL}\left(q_{g_{\phi(x)}}(z) \parallel p(z)\right)$$



Estimating the geometry

- **Statistical inconsistency:** $\hat{N}$ does not converge to N for $n \to +\infty$ if $\sigma \neq 0$

Proof: Counter-example.



- Toy PPCA model: $x_i = wz_i + \epsilon_i$ where $w \in \mathbb{R}$, x_i, z_i, ϵ_i have values in $\mathbb{R}$.
- Family: $Q = N(\beta, 1)$ to approximate $p_w(z|x)$.
- One layer linear decoder, one layer linear encoder.

$$\mathrm{ELBO}(\theta, \phi, x) = \log p_\theta(x) - \mathrm{KL}\left(q_{g_{\phi(x)}}(z|x) || p_\theta(z|x)\right)$$

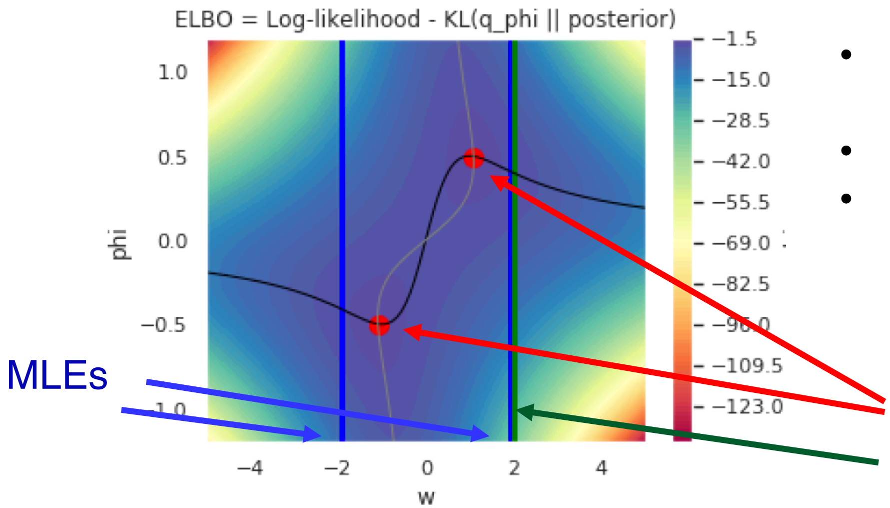
- **Geometric consistency:** $\hat{N} \to N$ for $\sigma \to 0$ and $n \to +\infty$

Proof: Computes the "geometric fit" to the manifold, known to be geometric consistent.

Estimating the geometry

- **Statistical inconsistency:** $\hat{N}$ does not converge to N for $n \to +\infty$ if $\sigma \neq 0$

Proof: Counter-example.



Maxima of ELBO

True value w = 2

- Toy PPCA model: $x_i = w z_i + \epsilon_i$ where $w \in \mathbb{R}$, x_i, z_i, ϵ_i have values in $\mathbb{R}$.
- Family: $Q = N(\beta, 1)$ to approximate $p_w(z|x)$.
- One layer linear decoder, one layer linear encoder.

$$\mathrm{ELBO}(\theta, \phi, x) = \log p_\theta(x) - \mathrm{KL}\left(q_{g_{\phi(x)}}(z\,|x)\,||\,p_\theta(z|x)\right)$$

- **Geometric consistency:** $\hat{N} \to N$ for $\sigma \to 0$ and $n \to +\infty$

Proof: Computes the "geometric fit" to the manifold, known to be geometric consistent.

- VAE estimating a flat curvature:
  - Either the submanifold is flat
  - Or the noise level is high: → no submanifold representing the data.

Outline: Learning submanifolds with gVAEs

Part 1

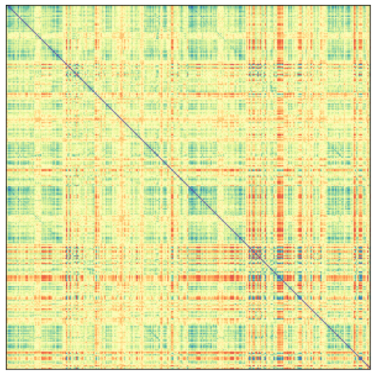
Probabilistic PCA,
Variational autoencoders
and manifold learning

Part 2

Geometric variational
autoencoders (gVAEs)
and submanifold learning

Part 3

Learning the submanifold
of functional brain
connectomes



- Geomstats: Implementing Riemannian Geometry and Geometric Statistics on GPUs
- Geometric Variational Autoencoders for functional connectomes

Geomstats

- **Geomstats:** Python package that gathers code from geometric statistics research into a shared unit-tested library, with backends enabling GPU computations.

```
pip3 install geomstats
export GEOMSTATS_BACKEND=numpy
```

Github repository: https://github.com/geomstats/geomstats
Documentation website: https://geomstats.github.io/
Contributing: https://geomstats.github.io/contributing.html + Hackathon in January.

Geomstats

- **Geomstats:** Python package that gathers code from geometric statistics research into a shared unit-tested library, with backends enabling GPU computations.

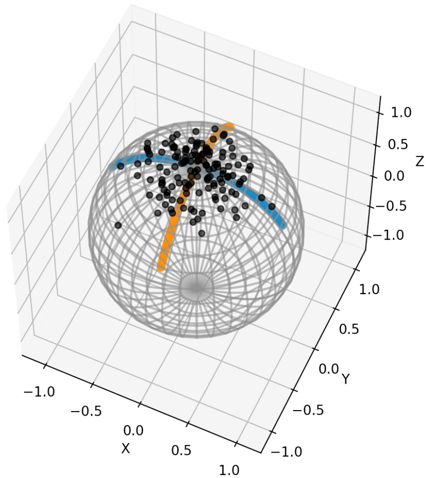
```
pip3 install geomstats
export GEOMSTATS_BACKEND=numpy
```

Github repository: https://github.com/geomstats/geomstats
Documentation website: https://geomstats.github.io/
Contributing: https://geomstats.github.io/contributing.html + Hackathon in January.

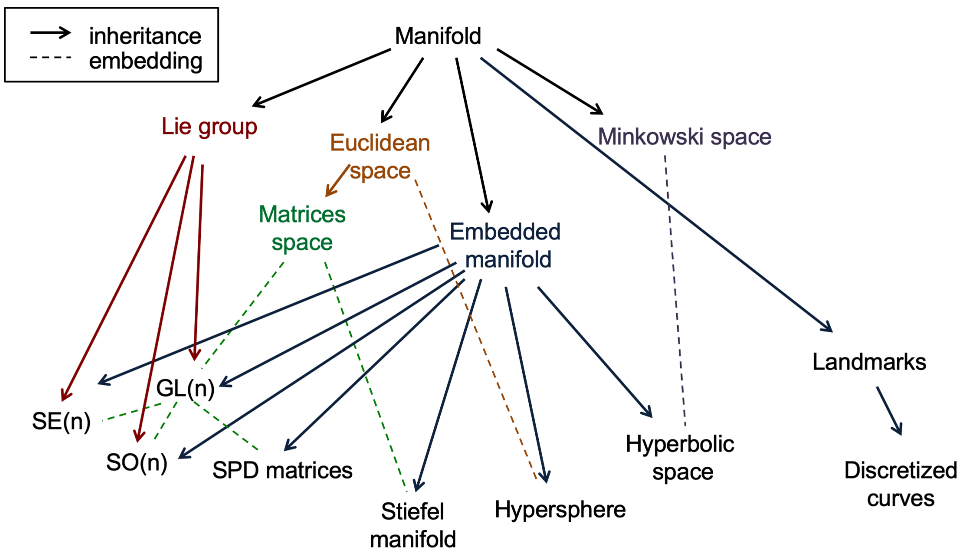
- Geomstats uses object-oriented programming (OOP) to implement two main modules: **geomstats.geometry** and **geomstats.learning**

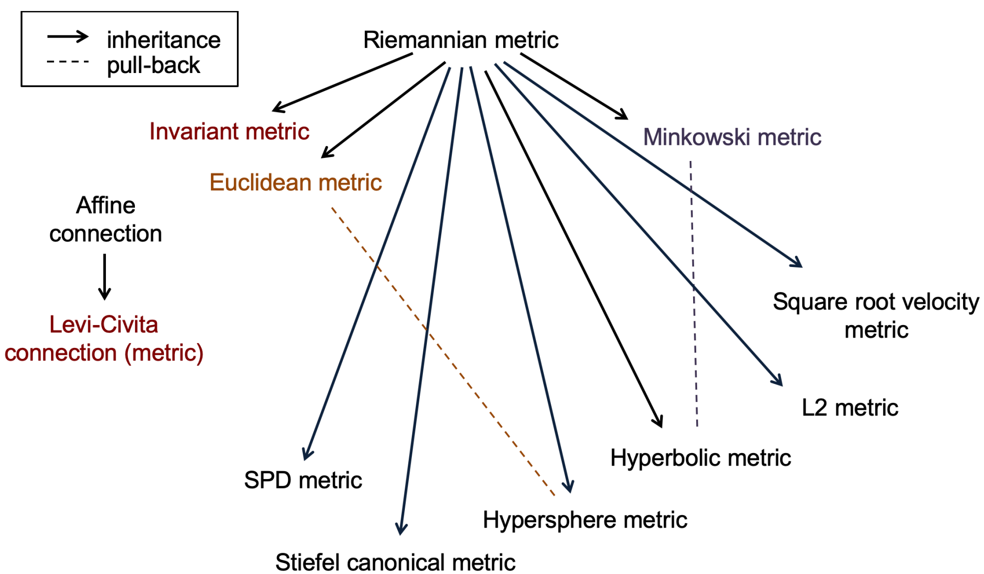
```
sphere = Hypersphere(dimension=2)
mean = sphere.metric.mean(data)

tpca = TangentPCA(metric=sphere.metric, n_components=2)
tpca = tpca.fit(data, base_point=mean)
```

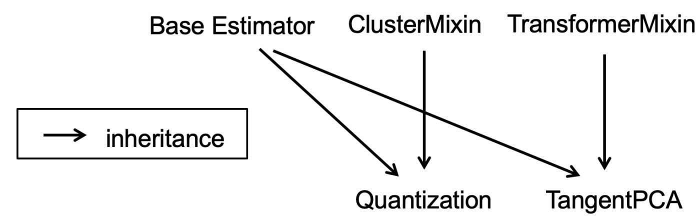


Geometry





Statistical Learning



- Geometry: mathematics API
- Learning: scikit-learn API

API: application program interface

A collaboration with: Pennec, Le Brigant, Mathe, Cabanes, Guigui, Thanwerdas, Kachan, Donnat, Jorda, et al.

Geomstats: Comparison with other libraries

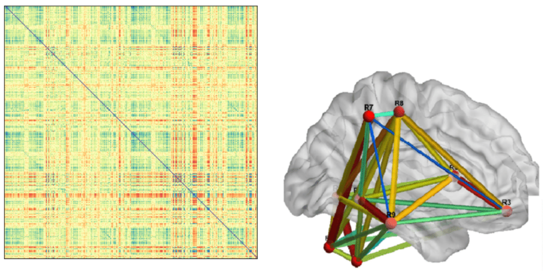
- Application specific:
    - **pyRiemann** (Barachant 2016)**:**
        - Riemannian geometry for covariance matrices

- Optimization:
    - **Pymanopt** (Townsend, Koep, Weichwald 2016):
        - Optimization on Riemannian manifolds
    - **McTorch** (Meghwanshi et al, 2018):
        - Optimization on Riemannian manifolds for deep learning
    - **Geoopt** (Becigneul, Ganea, Ferine, 2019):
        - Stochastic adaptive optimization on Riemannian manifolds

- Geometry focused:
    - **Theanogeometry** (Kuhnel, Sommer, 2017):
        - Non-linear statistics on manifolds of computational anatomy

Functional Brain Connectomes: Geometry?

fMRI



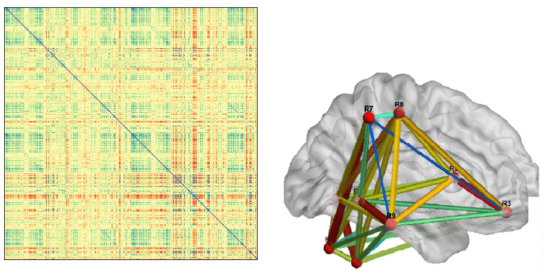
Brain connectomes data
(**Human Connectome Project)**

- 812 Subjects
- Correlations between brain areas

Measuring the dissimilarity between connectomes?

Functional Brain Connectomes: Geometry?

fMRI



Brain connectomes data
(**Human Connectome Project)**

- 812 Subjects
- Correlations between brain areas

Measuring the dissimilarity between connectomes?

Table 1: SPD matrix distances and their properties

Distance	Formula	Symmetric	Triangle inequality	Geodesic
Frobenius	$\|P_1 - P_2\|_F$	Yes	Yes	No
Cholesky-Frobenius [13]	$\|\mathrm{Chol}(P_1) - \mathrm{Chol}(P_2)\|_F$	Yes	Yes	No
J-divergence [12]	$\frac{1}{2}\sqrt{\mathrm{trace}(P_1 P_2^{-1} + P_2 P_1^{-1}) - 2n}$	Yes	No	No
Jensen-Bregman LogDet Divergence[11]	$\sqrt{\log\det\left(\frac{P_1+P_2}{2}\right) - \frac{1}{2}\log\det(P_1 P_2)}$	Yes	No	No
Affine-invariant [1]	$\|\log\left(P_1^{-1/2} P_2 P_1^{-1/2}\right)\|_F$	Yes	Yes	Yes
Log-Frobenius [6]	$\|\log(P_1) - \log(P_2)\|_F$	Yes	Yes	Yes

Table 2: SPD matrix distances and their properties

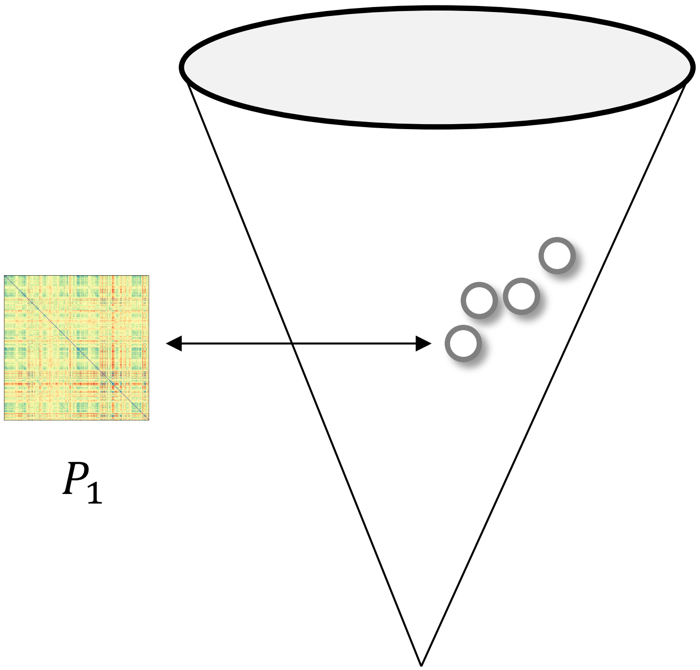
Distance	Distance from S_n^+	Affine invariance	Scale invariance	Rotation invariance	Inversion invariance
Frobenius	Finite	No	No	Yes	No
Cholesky-Frobenius [13]	Finite	No	No	No	No
J-divergence [12]	Infinite	Yes	Yes	Yes	Yes
Jensen-Bregman LogDet Divergence[11]	Infinite	Yes	Yes	Yes	Yes
Affine-invariant [1]	Infinite	Yes	Yes	Yes	Yes
Log-Frobenius [6]	Infinite	No	Yes	Yes	Yes

[Vemulapalli, Jacob, 2015][Donnat, Holmes, 2018][Thanwerdas, Pennec, 2019.]

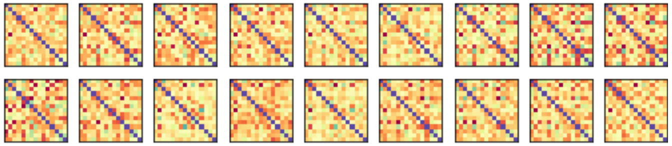
Functional Brain Connectomes: Geometry?

Affine invariant distance representing the dissimilarity between connectomes P1, P2:

$$d(P_1, P_2) = \left\| \log\left(P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}} \right) \right\|_{\text{Frob}}$$



One data point X_i, for $i = 1, \dots, n_{subjects}$
= One subject
= One connectome
= One SPD matrix of size $n_{\text{nodes}} \times n_{\text{nodes}}$
We choose $n_{\text{nodes}} = 15$. → D = 120



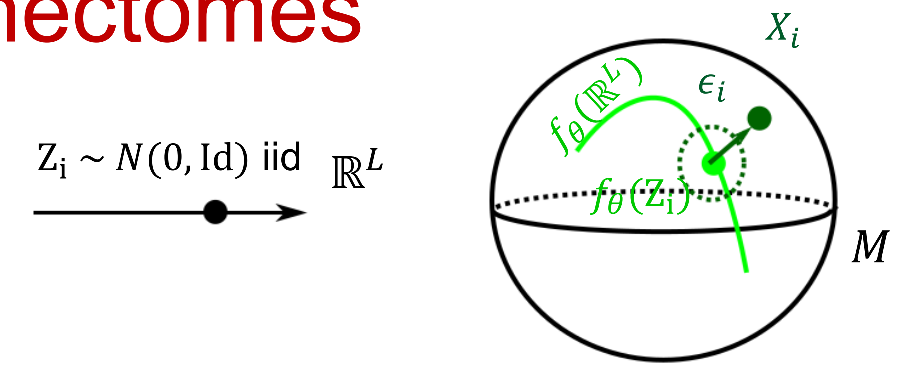
With this distance,

- Data space of connectomes is a cone.
- Cone borders are at infinite distance,
- Data on the borders correspond to null eigenvalues.

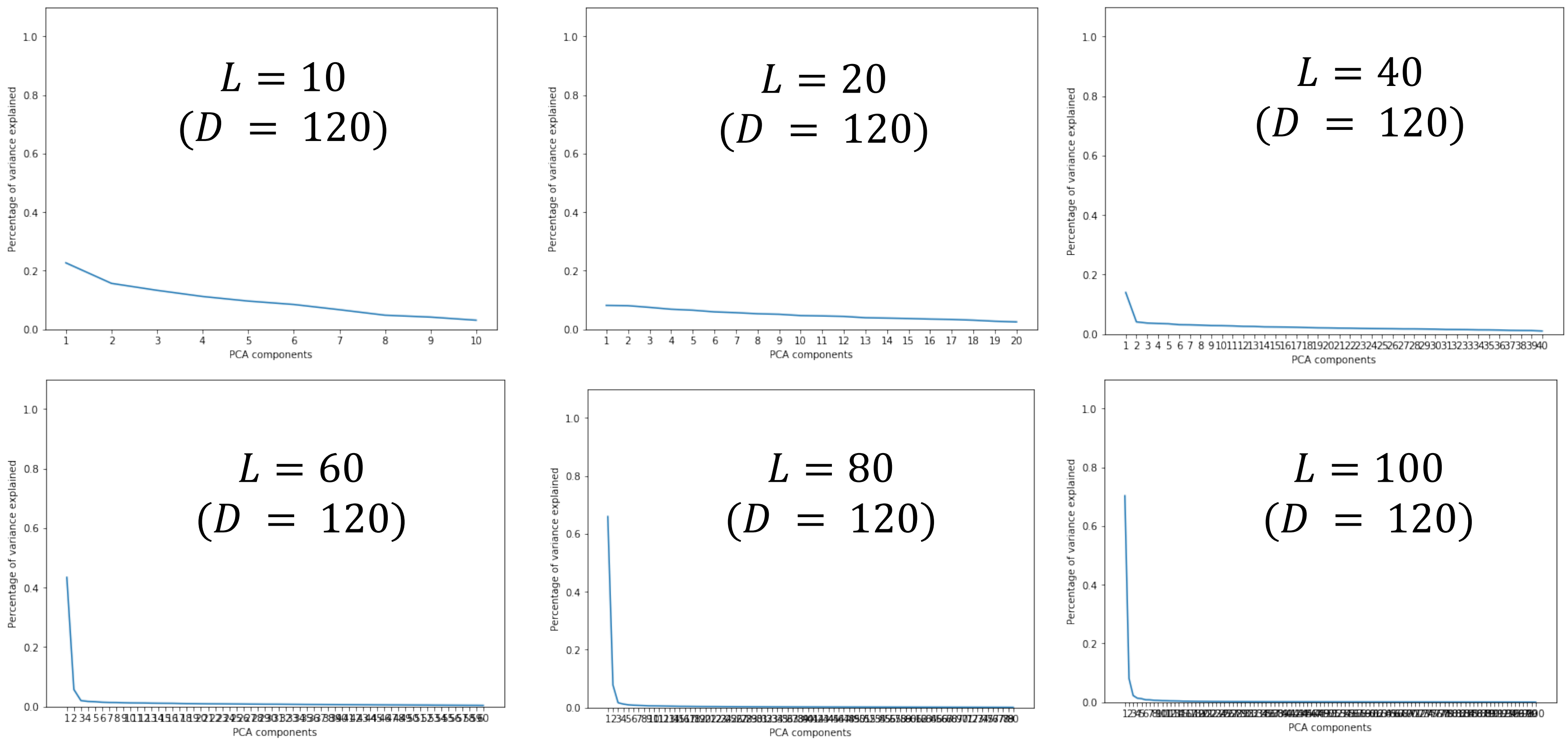
Geometric VAE for connectomes

If the differential $df_{\hat{\mu},\hat{W}}$ is of full rank:

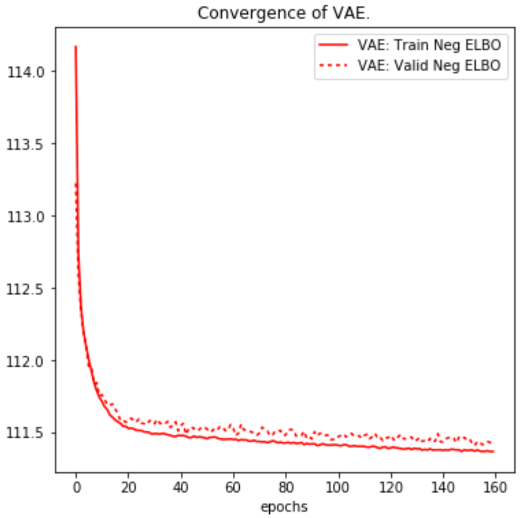
$\dim \hat{N} = L'$, dimension of the space spanned by the latent variables within $\mathbb{R}^L$.

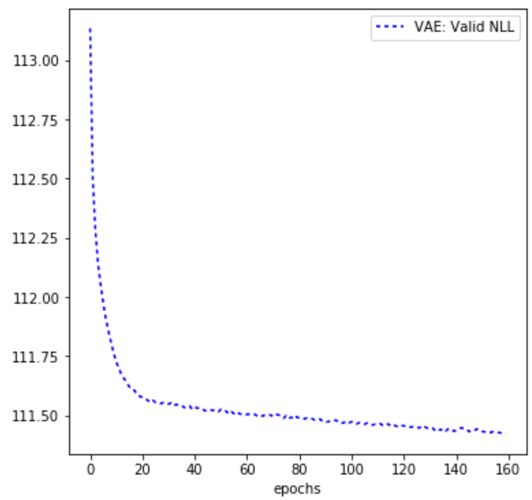


Percentage of variances explained:

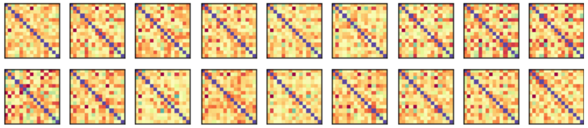


Geometric VAE for connectomes

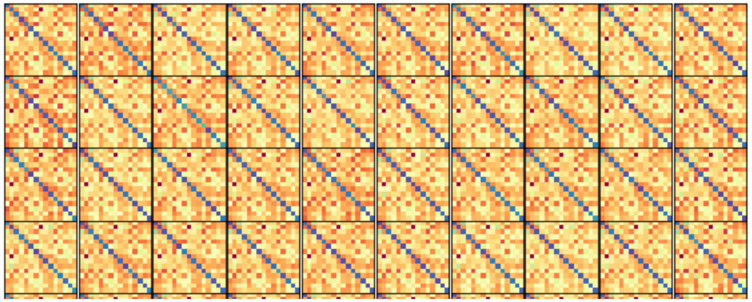




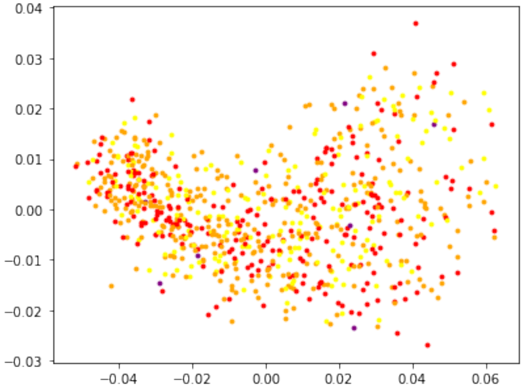
Data:



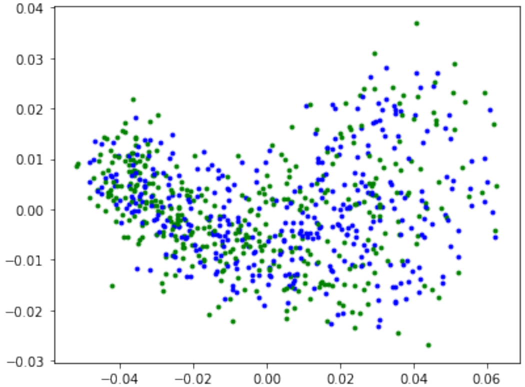
Generated data on submanifold $\widehat{N}$:



First 2 components in latent space $\mathbb{R}^{60}$: (~ 50% of variance)



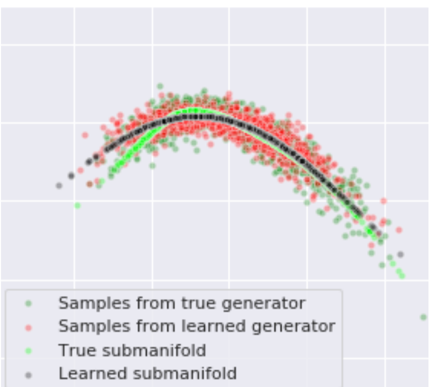
Colored by age group



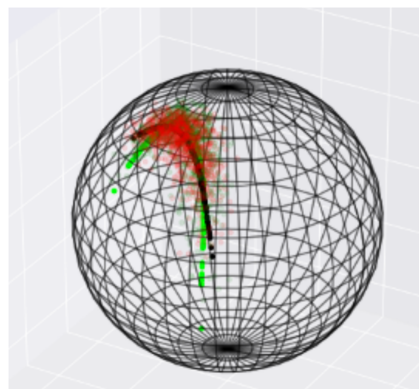
Colored by gender

Conclusion: learn submanifolds with gVAEs

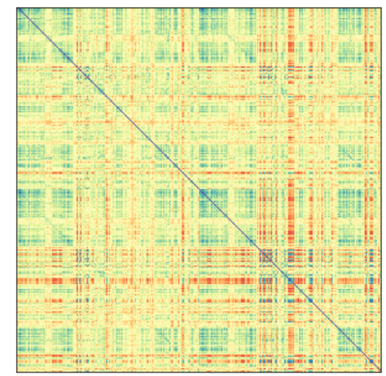
Probabilistic PCA,
Variational autoencoders
and manifold learning



Geometric variational
autoencoders (gVAEs)
and submanifold learning

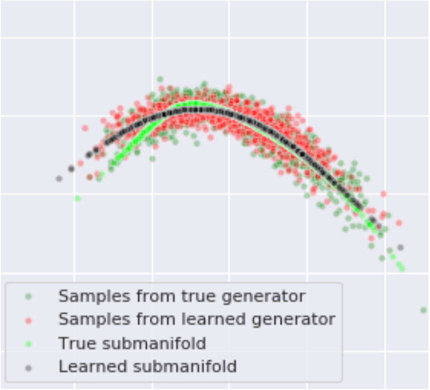


Learning the submanifold
of functional brain
connectomes

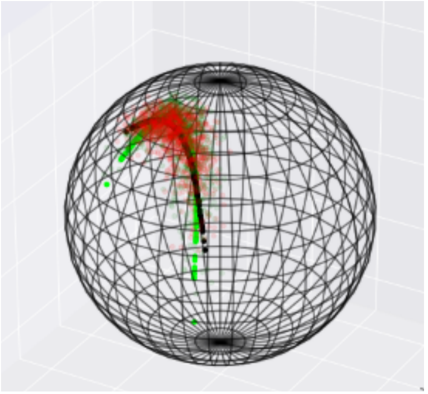


Conclusion: learn submanifolds with gVAEs

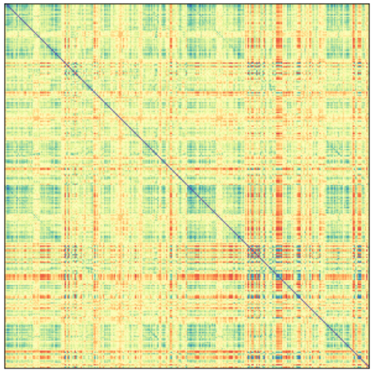
Probabilistic PCA, Variational autoencoders and manifold learning



Geometric variational autoencoders (gVAEs) and submanifold learning



Learning the submanifold of functional brain connectomes



Our question:

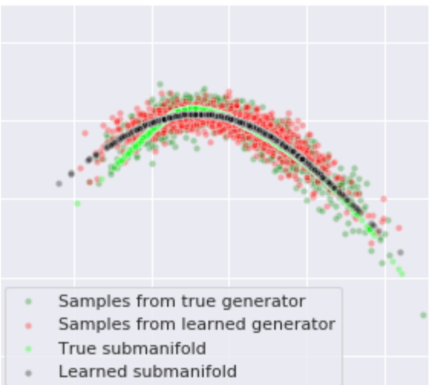
- **Can we extend traditional dimension reduction methods on Riemannian manifolds to learn "non-geodesic" submanifolds?**

Yes, we used:

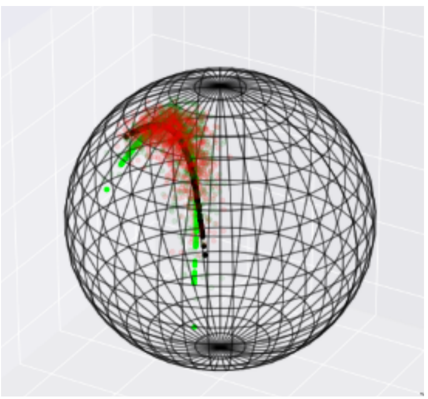
- Riemannian normal probability distributions for the model,
- Geomstats package for the implementation on GPUs.

Conclusion: learn submanifolds with gVAEs

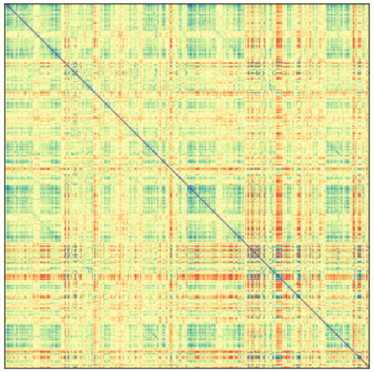
Probabilistic PCA, Variational autoencoders and manifold learning



Geometric variational autoencoders (gVAEs) and submanifold learning



Learning the submanifold of functional brain connectomes



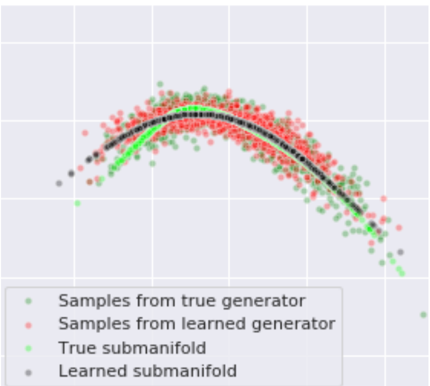
Our question:

- **What is the curvature of the learned submanifold: is it flat? [Fletcher 2014]**

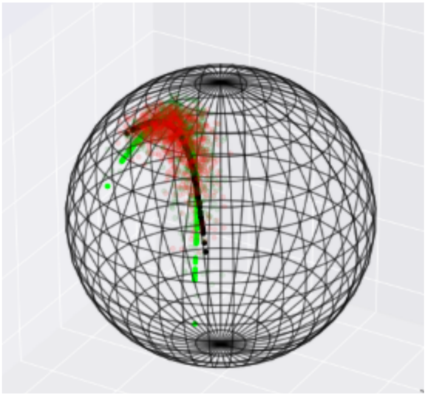
It seems more probable that the curvature is generally over-estimated, and this effect increases with the standard deviation of the noise.

Conclusion: learn submanifolds with gVAEs

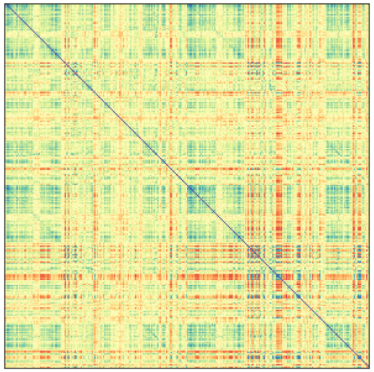
Probabilistic PCA, Variational autoencoders and manifold learning



Geometric variational autoencoders (gVAEs) and submanifold learning



Learning the submanifold of functional brain connectomes

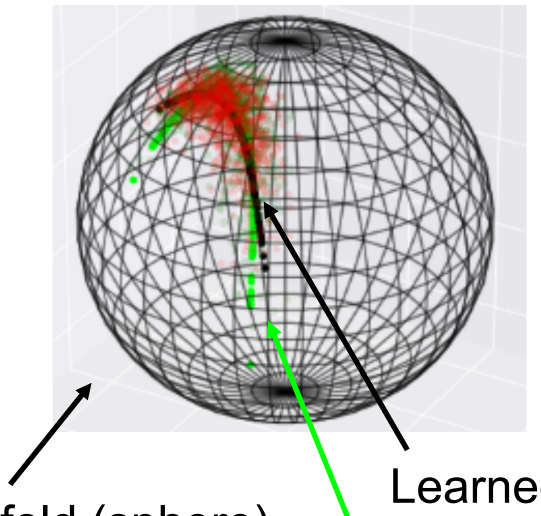


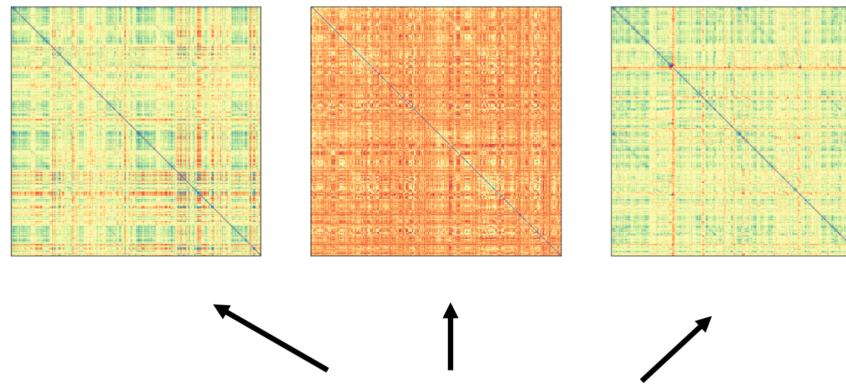
Our question:

- **Do more flexible models provide new insights on brain functional connectomes: are there patterns in resting state functional connectomes?**

We are able to detect patterns. We are discussing with functional neuroscience groups to compare with the literature and understand the patterns.

Learning submanifolds with geometric variational autoencoders: Application to brain functional connectomes





Thank you for your attention! Do you have questions?