Geometry-Aware Variational Autoencoders for Medical Data Augmentation.

Stéphanie Allassonnière*
join work with Clément Chadebec*
& Ninon Burgos† & Elina Thibeau-Sutre†

*Université Paris Cité - INRIA HeKA - INSERM
†INRIA Aramis - Institut du Cerveau - CNRS

Overview

1. Introduction
2. Variational Auto-Encoder - The Idea
   - Auto-Encoder
3. VAE framework
   - The idea
   - Mathematical foundations
   - Tweaking the approximate posterior distribution
4. Toward a Geometry-Aware VAE
   - The framework
   - The proposed model
   - A new way to generate data
   - Sensitivities and robustness on toy data
5. Results on Neuroimaging data
   - Materials
   - Methods
   - Results

Main Challenges

Main challenges with medical data

- Small number of subjects:
    - potential poor population representativity
    - no statistically significant results
    - overfitting

Main Challenges

Main challenges with medical data

- Small number of subjects:
  - potential poor population representativity
  - no statistically significant results
  - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Main Challenges

Main challenges with medical data

- Small number of subjects:
    - potential poor population representativity
    - no statistically significant results
    - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Need for

- Dimensionality reduction

Main Challenges

Main challenges with medical data

- Small number of subjects:
  - potential poor population representativity
  - no statistically significant results
  - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Need for

- Dimensionality reduction
  OR
- Data augmentation

Main Challenges

Main challenges with medical data

- Small number of subjects:
  - potential poor population representativity
  - no statistically significant results
  - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Need for

- Dimensionality reduction
  OR /AND
- Data augmentation

Main Challenges

Main challenges with medical data

- Small number of subjects:
  - potential poor population representativity
  - no statistically significant results
  - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Need for

- Dimensionality reduction
  OR /AND
- Data augmentation

A solution?

- Generative models: statistical hierarchical OR neural network based models

Main Challenges

Main challenges with medical data

- Small number of subjects:
  - potential poor population representativity
  - no statistically significant results
  - overfitting
- Large data (e.g. MRIs, omic data, etc..) $\Longrightarrow$ thousands of dimensions

Need for

- Dimensionality reduction
  OR /AND
- Data augmentation

A solution?

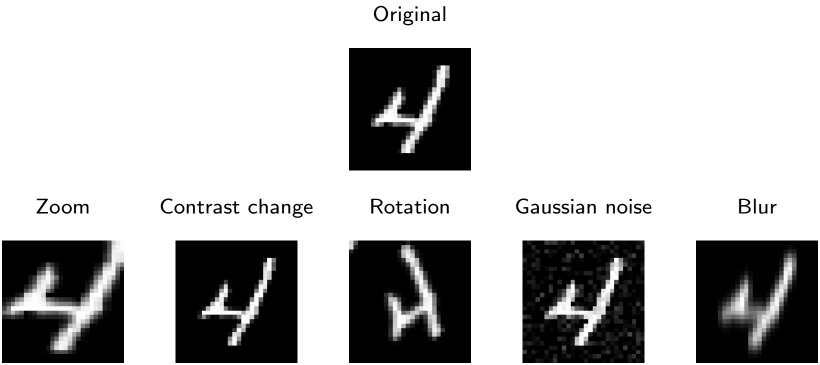
- Generative models: statistical hierarchical OR neural network based models

Issue

- Most of the time, unable to generate faithfully with small data sets

Classic Data Augmentation

- Adding some geometric transformations (shift, rotations ...)
- Adding noise, blur ...



Original

Zoom

Contrast change

Rotation

Gaussian noise

Blur

Figure: Examples of transformations

Classic Data Augmentation - Shortcomings

Classic DA

- Is data set dependent
- May require the intervention of an expert "knowledge"

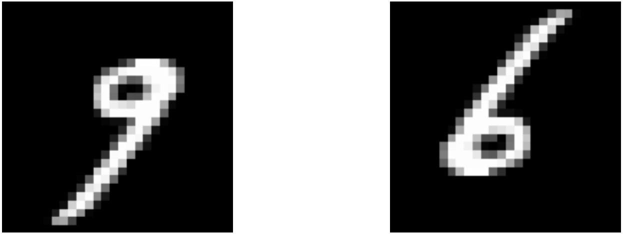


Figure: Nine figure rotated.

Classic Data Augmentation - Shortcomings

Classic DA

- Is data set dependent
- May require the intervention of an expert “knowledge”

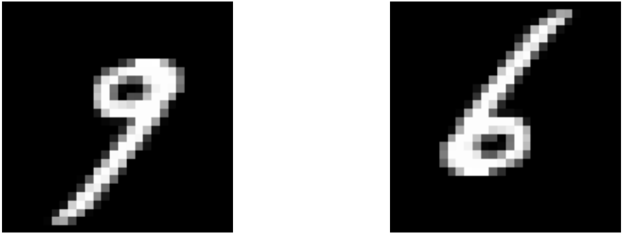


Figure: Nine figure rotated.

An attractive solution?

- Generative models (Generative Adversarial Networks, Variational Auto-Encoders ...)

Use of Generative Models for DA

GANs: wide use in many fields of application including medicine [YWB19]:

- Magnetic Resonance Images (MRI) [STR$^+$18, CMST17]
- Computed Tomography (CT) [FADK$^+$18, SYPS19]
- X-ray [MMKSM18, SVD$^+$18, WGG$^+$20],
- Positron Emission Tomography (PET) [BKK$^+$17],
- Mass spectroscopy data [LZL$^+$19],
- Dermoscopy [BAN18]
- Mammography [KRO$^+$18, WWCL18]

Use of Generative Models for DA

GANs: wide use in many fields of application including medicine [YWB19]:

- Magnetic Resonance Images (MRI) [STR$^+$18, CMST17]
- Computed Tomography (CT) [FADK$^+$18, SYPS19]
- X-ray [MMKSM18, SVD$^+$18, WGG$^+$20],
- Positron Emission Tomography (PET) [BKK$^+$17],
- Mass spectroscopy data [LZL$^+$19],
- Dermoscopy [BAN18]
- Mammography [KRO$^+$18, WWCL18]

$\Longrightarrow$ Most of these studies involved either a quite large training set (above 1000 training samples) or quite small dimensional data.

$\Longrightarrow$ As of today, the HDLSS setting remains poorly explored.

Use of Generative Models for DA

GANs: wide use in many fields of application including medicine [YWB19]:

- Magnetic Resonance Images (MRI) [STR$^+$18, CMST17]
- Computed Tomography (CT) [FADK$^+$18, SYPS19]
- X-ray [MMKSM18, SVD$^+$18, WGG$^+$20],
- Positron Emission Tomography (PET) [BKK$^+$17],
- Mass spectroscopy data [LZL$^+$19],
- Dermoscopy [BAN18]
- Mammography [KRO$^+$18, WWCL18]

$\Longrightarrow$ Most of these studies involved either a quite large training set (above 1000 training samples) or quite small dimensional data.

$\Longrightarrow$ As of today, the HDLSS setting remains poorly explored.

$\Longrightarrow$ Use VAEs!

Auto-Encoder

- The objective $\Longrightarrow$ Dimensionnality Reduction

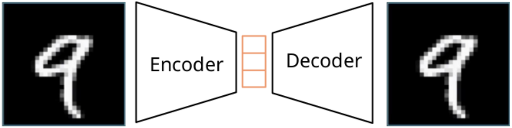


Figure: Simple Auto-Encoder

AutoEncoder

Assumptions:

- Let $x \in \mathcal{X}$ be a set a data. We assume that there exists $z \in \mathcal{Z}$ such that z is a low dimensional representation of x
- The encoder e_θ and decoder d_ϕ are functions modelled by neural networks (NNs) such that θ and ϕ are the weights of the NNs
- Let x' be the reconstructed samples, the objective is to have $x \simeq x'$

The Objective function writes:

$$\mathcal{L} = \|x - x'\|^2 = \|x - d_\phi(z)\|^2 = \|x - d_\phi(e_\theta(x))\|^2$$

AutoEncoder

Assumptions:

- Let $x \in \mathcal{X}$ be a set a data. We assume that there exists $z \in \mathcal{Z}$ such that z is a low dimensional representation of x
- The encoder e_θ and decoder d_ϕ are functions modelled by neural networks (NNs) such that θ and ϕ are the weights of the NNs
- Let x' be the reconstructed samples, the objective is to have $x \simeq x'$

The Objective function writes:

$$\mathcal{L} = \|x - x'\|^2 = \|x - d_\phi(z)\|^2 = \|x - d_\phi(e_\theta(x))\|^2$$

$\Longrightarrow$ The networks are optimised using stochastic gradient descent

$$\phi \leftarrow \phi - \varepsilon \cdot \nabla_\phi \mathcal{L}$$
$$\theta \leftarrow \theta - \varepsilon \cdot \nabla_\theta \mathcal{L}$$

AutoEncoder - Shortcomings

- How to generate new data ?

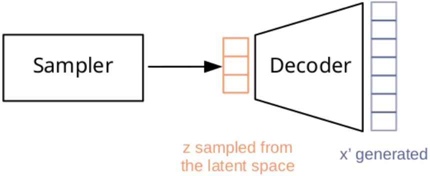


Figure: Generation procedure ?

AutoEncoder - Shortcomings

- How to generate new data ?

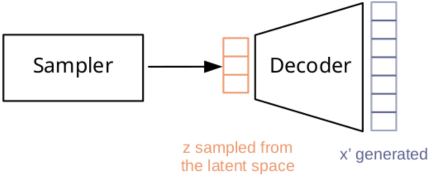


Figure: Generation procedure ?

- How to sample form the latent space?

AutoEncoder - Shortcomings

- How to generate new data ?

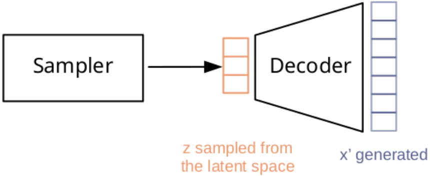


Figure: Generation procedure ?

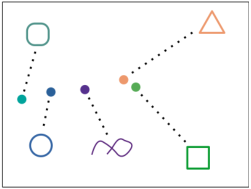


Figure: Potential latent space

- How to sample form the latent space?
- The AutoEncoder was just trained to encode and decode the **input data** without information on its structure or distribution.

AutoEncoder - Shortcomings

- How to generate new data ?

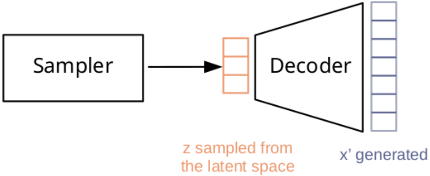


Figure: Generation procedure ?

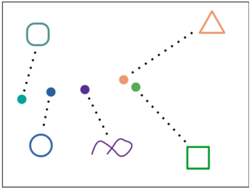


Figure: Potential latent space

- How to sample form the latent space?
- The AutoEncoder was just trained to encode and decode the **input data** without information on its structure or distribution.

$\Longrightarrow$ Need for a new framework

VAE - The Idea

- An auto-encoder based model...

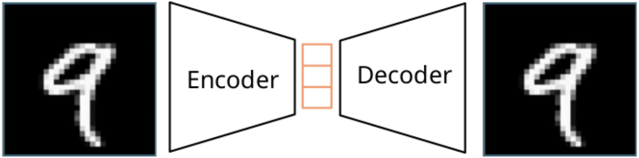


Figure: Simple Auto-Encoder

- ... but where an input data point is encoded as a **distribution** defined over the latent space [KW14, RMW14]

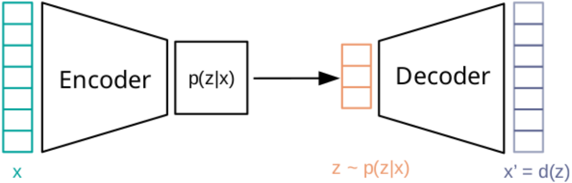


Figure: VAE framework

VAE - Mathematical Considerations

- Let $x \in \mathcal{X}$ be a set of data and $\{P_\theta, \theta \in \Theta\}$ be a parametric model
- We assume there exists latent variables $z \in \mathcal{Z}$ living in a smaller space such that the marginal likelihood writes

$$p_\theta(x) = \int p_\theta(x|z) q_{\mathrm{prior}}(z) dz \, ,$$

where q_{prior} is a prior distribution over the latent variables and $p_\theta(x|z)$ is referred to as the decoder

VAE - Mathematical Considerations

- Let $x \in \mathcal{X}$ be a set of data and $\{P_\theta, \theta \in \Theta\}$ be a parametric model
- We assume there exists latent variables $z \in \mathcal{Z}$ living in a smaller space such that the marginal likelihood writes

  $$p_\theta(x) = \int p_\theta(x|z) q_{\mathrm{prior}}(z) dz \,,$$

  where q_{prior} is a prior distribution over the latent variables and $p_\theta(x|z)$ is referred to as the decoder
- Example:

  $$q_{\mathrm{prior}} = \mathcal{N}(0, I), \qquad p_\theta(x|z) = \prod_{i=1}^{D} \mathcal{B}(\pi_{\theta_i(z)})$$

VAE - Mathematical Considerations

- Let $x \in \mathcal{X}$ be a set of data and $\{P_\theta, \theta \in \Theta\}$ be a parametric model
- We assume there exists latent variables $z \in \mathcal{Z}$ living in a smaller space such that the marginal likelihood writes

$$p_\theta(x) = \int p_\theta(x|z) q_{\mathrm{prior}}(z) dz \, ,$$

  where q_{prior} is a prior distribution over the latent variables and $p_\theta(x|z)$ is referred to as the decoder
- Example:

$$q_{\mathrm{prior}} = \mathcal{N}(0, I), \qquad p_\theta(x|z) = \prod_{i=1}^{D} \mathcal{B}(\pi_{\theta_i(z)})$$

Objective:

- Maximizing the likelihood of the model

VAE - Mathematical Considerations

- Let $x \in \mathcal{X}$ be a set of data and $\{P_\theta, \theta \in \Theta\}$ be a parametric model
- We assume there exists latent variables $z \in \mathcal{Z}$ living in a smaller space such that the marginal likelihood writes

$$p_\theta(x) = \int p_\theta(x|z) q_{\mathrm{prior}}(z) dz\,,$$

  where q_{prior} is a prior distribution over the latent variables and $p_\theta(x|z)$ is referred to as the decoder

- Example:

$$q_{\mathrm{prior}} = \mathcal{N}(0, I), \qquad p_\theta(x|z) = \prod_{i=1}^{D} \mathcal{B}(\pi_{\theta_i(z)})$$

Objective:

- Maximizing the likelihood of the model

Problem: The integral is often intractable.

Variational inference

We have to use Variational Inference:

$$\begin{aligned}
\log p_\theta(x) &= \log\left(\int p_\theta(x|z) q_{\mathrm{prior}}(z) dz\right) \\
&= \log\left(\int p_\theta(x,z) dz\right) \\
&= \log\left(\int p_\theta(x,z) \frac{q(z)}{q(z)} dz\right)\,, \text{ for any pdf } q \\
&\geq \int \left(\log \frac{p_\theta(x,z)}{q(z)}\right) q(z) dz\,, \text{ using Jensen's inequality} \\
&\geq \int \left(\log p_\theta(x,z)\right) q(z) dz - H(q(z))
\end{aligned}$$

with H the entropy of $q(z)$.

The equality holds for $q(z) = q_\theta(z|x)$.

Variational inference: The ELBO

- Well-know issue: the posterior $q(z) = q_\theta(z|x)$ is intractable.
  $\longrightarrow$ use Expectation-Maximization like algorithms (MCMC-SAEM version if needed)
- **OR** approximate this posterior:

Variational inference: The ELBO

- Well-know issue: the posterior $q(z) = q_\theta(z|x)$ is intractable.

  $\longrightarrow$ use Expectation-Maximization like algorithms (MCMC-SAEM version if needed)
- **OR** approximate this posterior:

  Introduce a parametric approximation:

$$q_\phi(z|x) \simeq p_\theta(z|x)\,,$$

where for example $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$

Variational inference: The ELBO

- Well-know issue: the posterior $q(z) = q_\theta(z|x)$ is intractable.
  $\longrightarrow$ use Expectation-Maximization like algorithms (MCMC-SAEM version if needed)
- **OR** approximate this posterior:
  Introduce a parametric approximation:

$$q_\phi(z|x) \simeq p_\theta(z|x)\,,$$

where for example $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$
This leads to an unbiased estimate of the log-likelihood

$$\widehat{p_\theta}(x) = \frac{p_\theta(x,z)}{q_\phi(z|x)}, \qquad \mathbb{E}_{z\sim q_\phi(z|x)}[\widehat{p_\theta}(x)] = p_\theta(x)\,,$$

Variational inference: The ELBO

- Well-know issue: the posterior $q(z) = q_\theta(z|x)$ is intractable.
  $\longrightarrow$ use Expectation-Maximization like algorithms (MCMC-SAEM version if needed)
- **OR** approximate this posterior:
  Introduce a parametric approximation:

$$q_\phi(z|x) \simeq p_\theta(z|x)\,,$$

where for example $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$
This leads to an unbiased estimate of the log-likelihood

$$\widehat{p}_\theta(x) = \frac{p_\theta(x,z)}{q_\phi(z|x)}, \qquad \mathbb{E}_{z\sim q_\phi(z|x)}[\widehat{p}_\theta(x)] = p_\theta(x)\,,$$

and the definition of the **Evidence Lower Bound** (ELBO):

$$\begin{aligned} \log p_\theta(x) &\geq \mathbb{E}_{z\sim q_\phi(z|x)}[\log(p_\theta(x,z)) - \log(q_\phi(z|x))] \\ &\geq ELBO \end{aligned}$$

Variational inference: The ELBO

Objectives:

1. Optimize the ELBO **as a function** instead of the target distribution

   Use stochastic gradient descent in both θ and ϕ

Variational inference: The ELBO

Objectives:

1. Optimize the ELBO **as a function** instead of the target distribution

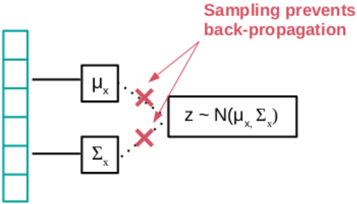
   Use stochastic gradient descent in both θ and ϕ

2. Optimize the ELBO **as a bound** to get closer to the target

   Use sampling methods to produce samples $z \sim q_\theta(z|x)$

The Reparametrization Trick for stochastic gradient descent

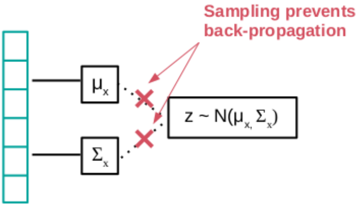
- Since $z \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, the model is not amenable to gradient descent


Sampling prevents back-propagation
μ_x
Σ_x
$z \sim N(\mu_x, \Sigma_x)$

(a) Back-propagation impossible

The Reparametrization Trick for stochastic gradient descent

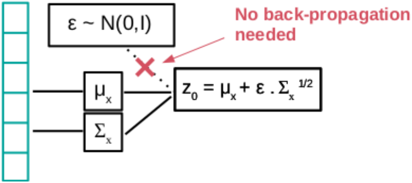
- Since $z \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, the model is not amenable to gradient descent



Sampling prevents back-propagation

(a) Back-propagation impossible

$\Longrightarrow$

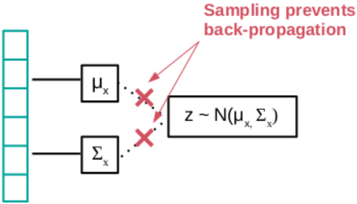

$\varepsilon \sim N(0,I)$
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \,.\, \Sigma_x^{1/2}$

No back-propagation needed

(b) Back-propagation possible: samples are differentiable functions of the parameters

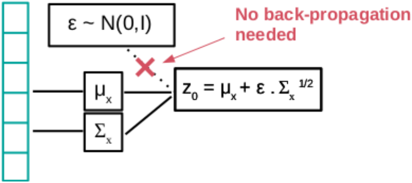
The Reparametrization Trick for stochastic gradient descent

- Since $z \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, the model is not amenable to gradient descent


Sampling prevents back-propagation
μ_x
Σ_x
$z \sim N(\mu_x, \Sigma_x)$

(a) Back-propagation impossible

$\Longrightarrow$


$\varepsilon \sim N(0,I)$
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \,.\, \Sigma_x^{1/2}$

(b) Back-propagation possible: samples are differentiable functions of the parameters

$\Longrightarrow$ Optimization with respect to encoder and decoder parameters made possible !

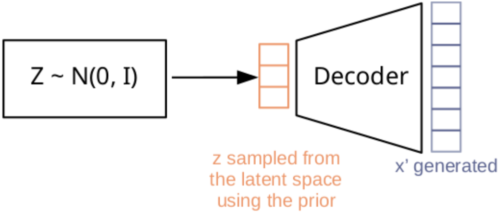
Objective 1.

Generating new samples

Back to the model:

$$p_\theta(x) = \int p_\theta(x|z) q_{\text{prior}}(z) dz \,,$$

- We only need to sample $z \sim q_{\text{prior}}(z) = \mathcal{N}(0, I)$ and feed it to the decoder.

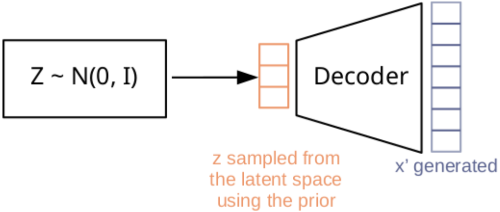


Generating new samples

Back to the model:

$$p_\theta(x) = \int p_\theta(x|z) q_{\text{prior}}(z) dz \,,$$

- We only need to sample $z \sim q_{\text{prior}}(z) = \mathcal{N}(0, I)$ and feed it to the decoder.



Pros:

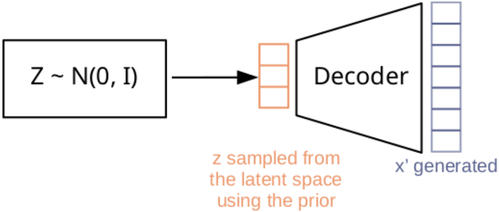
- Very simple to use in practice

Generating new samples

Back to the model:

$$p_\theta(x) = \int p_\theta(x|z) q_{\text{prior}}(z) dz \, ,$$

- We only need to sample $z \sim \; q_{\text{prior}}(z) \; = \mathcal{N}(0, I)$ and feed it to the decoder.



Pros:

- Very simple to use in practice

Cons:

- The prior and posterior are not expressive enough to capture complex distributions
- Poor latent space prospecting

Tweaking the Approximate Posterior Distribution

Concerning Objective 2.

- The ELBO can written as

$$ELBO = \log p_\theta(x) - \underbrace{\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))}_{\approx 0 \text{ if } q_\phi(z|x) \approx p_\theta(z|x)} .$$

- Kullback-Leiber divergence $\geq 0 \Rightarrow$ make it vanish by tweaking the approximate posterior $q_\phi(z|x)$

- Produce variables z which targets the true posterior $p_\theta(z|x)$ using a sample $z_0 \sim q_{init}$

Tweaking the Approximate Posterior Distribution

Concerning Objective 2.

- The ELBO can written as

$$ELBO = \log p_\theta(x) - \underbrace{\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))}_{\approx 0 \text{ if } q_\phi(z|x) \approx p_\theta(z|x)} .$$

- Kullback-Leiber divergence $\geq 0 \Rightarrow$ make it vanish by tweaking the approximate posterior $q_\phi(z|x)$

- Produce variables z which targets the true posterior $p_\theta(z|x)$ using a sample $z_0 \sim q_{init}$

- How? and how to ensure that the model would still be amenable to the back-propagation ?

Solution 1: Normalizing Flows

- Use smooth invertible parametrized mappings f_ψ to “sample” z [RM15]

- Apply K transformations to $z_0 \sim q_{init}$ (here $q_{init} = q_\phi$)
- Final random variable $z_K = f_x^K \circ \cdots \circ f_x^1(z_0) \sim q_\phi(z_K|x)$ with

$$q_\phi(z_K|x) = q_\phi(z_0|x) \prod_{k=1}^{K} |\det \mathbf{J}_{f_x^k}|^{-1}, \tag{1}$$

Solution 1: Normalizing Flows

- Use smooth invertible parametrized mappings f_ψ to “sample” z [RM15]

- Apply K transformations to $z_0 \sim q_{init}$ (here $q_{init} = q_\phi$)
- Final random variable $z_K = f_x^K \circ \cdots \circ f_x^1(z_0) \sim q_\phi(z_K|x)$ with

$$q_\phi(z_K|x) = q_\phi(z_0|x) \prod_{k=1}^{K} |\det \mathbf{J}_{f_x^k}|^{-1}, \tag{1}$$

Objective 2.

✓

although difficult to compute the Jacobian of these maps f_1^K

Solution 2: Hamiltonian VAE

- Idea = Hybrid Monte Carlo Sampler [No11, DMS17, LBB+19],
- Target density

$$p_\theta(z|x) = \frac{p_\theta(x,z)}{p_\theta(x)} \propto p_\theta(x,z) = \pi_x(z)\,.$$

- Introduce an auxiliary random variable $\rho \sim \mathcal{N}(0, \mathbf{M})$ called "momentum"
- Write the Hamiltonian:

$$\begin{aligned} H_x(z,\rho) &= -\log \pi_x(z,\rho) \\ &= -\log \pi_x(z) + \frac{1}{2}\log((2\pi)^d|\mathbf{M}|) + \rho^\top \mathbf{M}^{-1}\rho \\ &= U_x(z) + \kappa(\rho)\,. \end{aligned}$$

- Sample (z,ρ) with this dynamic.

Solution 2: Hamiltonian VAE

- Use a discretization scheme

$$\begin{aligned} \rho(t+\varepsilon/2) &= \rho(t) - \frac{\varepsilon}{2} \cdot \nabla_z H(z(t), \rho(t))\,, \\ z(t+\varepsilon) &= z(t) + \varepsilon \cdot \nabla_\rho (H(z(t), \rho(t+\varepsilon/2)))\,, \\ \rho(t+\varepsilon) &= \rho(t+\varepsilon/2) - \frac{\varepsilon}{2} \cdot \nabla_z H(z(t+\varepsilon), \rho(t+\varepsilon/2))\,, \end{aligned} \tag{2}$$

- A proposal $(\widetilde{z}, \widetilde{\rho})$ is accepted with probability:

$$\alpha = \min\Big(1, \exp\big(-H(\widetilde{z}, \widetilde{\rho}) + H(z, \rho)\big)\Big)$$

$\Longrightarrow$ Creates an ergodic, time-reversible Markov Chain having π_x as stationary distribution.

Note that the Metropolis Hastings' acceptation step has to be removed for the back propagation to be possible.

Hamiltonian VAE

- The graphical scheme [CDS18]

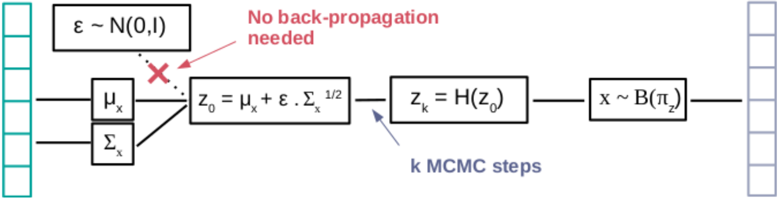

ε ~ N(0,I)
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \, . \, \Sigma_x^{1/2}$
$z_k = H(z_0)$
$x \sim B(\pi_z)$
k MCMC steps

Figure: Hamiltonian VAE

Hamiltonian VAE

- The graphical scheme [CDS18]

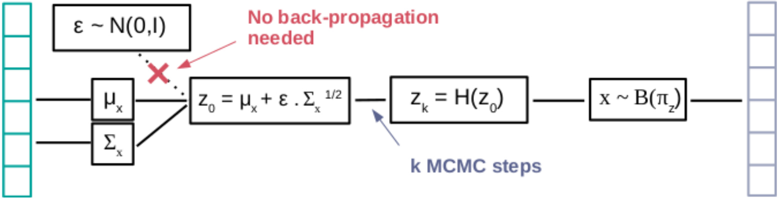

ε ~ N(0,I)
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \, . \, \Sigma_x^{1/2}$
$z_k = H(z_0)$
k MCMC steps
$x \sim B(\pi_z)$

Figure: Hamiltonian VAE

Issue: Perform poorly when trained on small data set and so we need to define a new framework

Hamiltonian VAE

- The graphical scheme [CDS18]

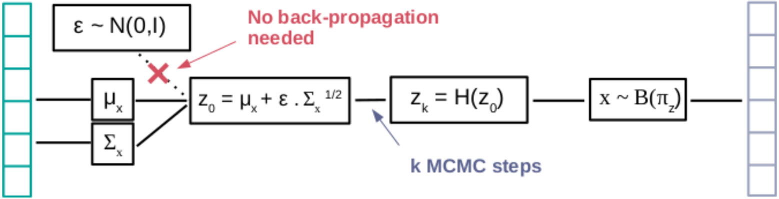

ε ~ N(0,I)
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \,.\, \Sigma_x^{1/2}$
$z_k = H(z_0)$
$x \sim B(\pi_z)$
k MCMC steps

Figure: Hamiltonian VAE

Issue: Perform poorly when trained on small data set and so we need to define a new framework

Hamiltonian VAE

- The graphical scheme [CDS18]

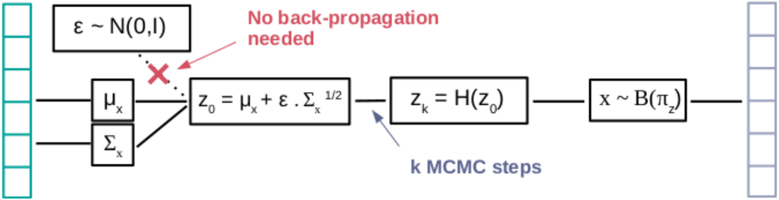

$\varepsilon \sim N(0,I)$
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon . \Sigma_x^{1/2}$
$z_k = H(z_0)$
$x \sim B(\pi_z)$
k MCMC steps

Figure: Hamiltonian VAE

Issue: Perform poorly when trained on small data set and so we need to define a new framework

What about geometry?

Defining a New Framework

Assumptions:

- As of now the latent space structure was supposed to be Euclidean (i.e. $\mathcal{Z} = \mathbb{R}^d$)
- Let us now relax this hypothesis and assume that $\mathcal{Z}$ is a Riemannian manifold endowed with a metric $\mathbf{G}$.
- It was shown that exploiting the geometrical aspect of probability distributions can lead to far more efficient sampling [GCC09, GC11]

Defining a New Framework

Assumptions:

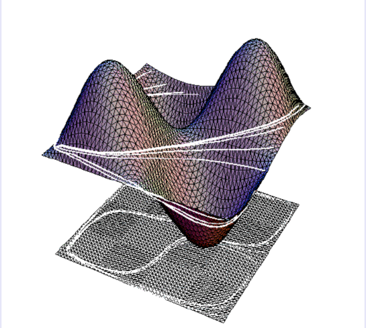
- As of now the latent space structure was supposed to be Euclidean (i.e. $\mathcal{Z} = \mathbb{R}^d$)
- Let us now relax this hypothesis and assume that $\mathcal{Z}$ is a Riemannian manifold endowed with a metric $\mathbf{G}$.
- It was shown that exploiting the geometrical aspect of probability distributions can lead to far more efficient sampling [GCC09, GC11]

Our ideas:

1. Exploit the manifold structure of the latent space to improve the posterior sampling
2. Learn the metric defined in the latent space
3. Use the learned geometry to generate instead of the prior [CTSBA21]

Riemanian geometry principles

- Riemanian manifold: (reduced to our model) $\mathbb{R}^d$ endowed with a metric $\mathbf{G}$: $\mathcal{M} = (\mathbb{R}^d, \mathbf{G})$.
  $\Longrightarrow \mathbb{R}^d$ not flat anymore, curved space (as montains)



Shortest path geodesic on sinusoidal surface. See Ref. 1 below.

Figure: Image taken from: Fast Marching Methods on Triangulated Domains : Kimmel, R., a Sethian, J.A., Proceedings of the National Academy of Sciences, 95, pp. 8341-8435, 1998

Riemanian geometry principles

- Riemanian manifold: (reduced to our model) $\mathbb{R}^d$ endowed with a metric $\mathbf{G}$: $\mathcal{M} = (\mathbb{R}^d, \mathbf{G})$.
  $\Longrightarrow \mathbb{R}^d$ not flat anymore, curved space (as montains)
- Geodesic curves:
  - Length of a curve $\gamma : [0, 1] \to \mathcal{M}$ from z_1 to z_2 living in a Riemannian manifold $\mathcal{M}$

$$L(\gamma) = \int_0^1 \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)}} dt \qquad \gamma(0) = z_1, \gamma(1) = z_2 \,. \tag{3}$$

  - Geodesic paths = curve γ minimizing Eq. (3)
  - or equivalently minimizing the curve energy

$$E(\gamma) = \int_0^1 \langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)} dt \qquad \gamma(0) = z_1, \gamma(1) = z_2 \,.$$

1) Improve Posterior Sampling - Riemannian HMC

- Rely on the Riemannian Hamiltonian Monte Carlo Sampler [GC11]:
    - Introduce a Position-specific random momentum $\rho \sim \mathcal{N}(0, \mathbf{G}(z))$
    - Simulates the evolution $(z(t), \rho(t))$ of a particle whose motion is governed by Hamiltonian dynamics on the manifold
- The Hamiltonian writes

$$H_x^{Riem}(z, \rho) = \log p_{\mathrm{target}}(z) + \frac{1}{2} \log((2\pi)^D \det \mathbf{G}(z)) + \frac{1}{2} \rho^\top \mathbf{G}(z)^{-1} \rho \,.$$

1) Improve Posterior Sampling - Riemannian HMC

- Rely on the Riemannian Hamiltonian Monte Carlo Sampler [GC11]:
  - Introduce a Position-specific random momentum $\rho \sim \mathcal{N}(0, \mathbf{G}(z))$
  - Simulates the evolution $(z(t), \rho(t))$ of a particle whose motion is governed by Hamiltonian dynamics on the manifold
- The Hamiltonian writes

$$H_x^{Riem}(z,\rho) = \log p_{\mathrm{target}}(z) + \frac{1}{2}\log((2\pi)^D \det \mathbf{G}(z)) + \frac{1}{2}\rho^\top \mathbf{G}(z)^{-1}\rho\,.$$

- Use of the "Generalized" Leapfrog integrator to sample from p_{target}

1) Improve Posterior Sampling - Riemannian HMC

- Rely on the Riemannian Hamiltonian Monte Carlo Sampler [GC11]:
  - Introduce a Position-specific random momentum $\rho \sim \mathcal{N}(0, \mathbf{G}(z))$
  - Simulates the evolution $(z(t), \rho(t))$ of a particle whose motion is governed by Hamiltonian dynamics on the manifold
- The Hamiltonian writes

$$H_x^{Riem}(z, \rho) = \log p_{\mathrm{target}}(z) + \frac{1}{2} \log((2\pi)^D \det \mathbf{G}(z)) + \frac{1}{2}\rho^\top \mathbf{G}(z)^{-1} \rho \,.$$

- Use of the "Generalized" Leapfrog integrator to sample from p_{target}

Pros:

- Use the underlying geometry of the data to improve sampling

Cons:

- The metric is unknown

2) Learn the Metric - The Choice of the Metric

- Parametric metric: [Lou19]:

$$\mathbf{G}^{-1}(z) = \sum_{i=1}^{N} L_{\psi_i} L_{\psi_i}^{\top} \exp\Big(- \frac{\|z - c_i\|_2^2}{T^2} \Big) + \lambda I_d \ ,$$

- L_{ψ_i} lower triangular matrices parametrized using neural networks
- T temperature to smooth the metric
- c_i centroids
- λ regularization factor

2) Learn the Metric - The Choice of the Metric

- Parametric metric: [Lou19]:

$$\mathbf{G}^{-1}(z) = \sum_{i=1}^{N} L_{\psi_i} L_{\psi_i}^{\top} \exp\Big(-\frac{\|z - c_i\|_2^2}{T^2}\Big) + \lambda I_d ,$$

- L_{ψ_i} lower triangular matrices parametrized using neural networks
- T temperature to smooth the metric
- c_i centroids
- λ regularization factor

Pros:

- Closed-form expression of the inverse metric $\Longrightarrow$ useful for geodesic computation
- Geodesics travel through most populated areas.

The Model - Riemannian Hamiltonian VAE

- The graphical scheme

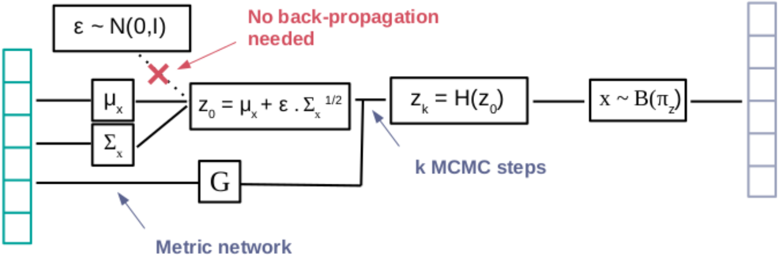
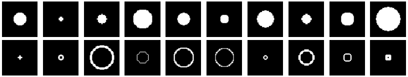

$\varepsilon \sim N(0,I)$
No back-propagation needed
μ_x
Σ_x
$z_0 = \mu_x + \varepsilon \cdot \Sigma_x^{1/2}$
$z_k = H(z_0)$
$x \sim B(\pi_z)$
G
k MCMC steps
Metric network

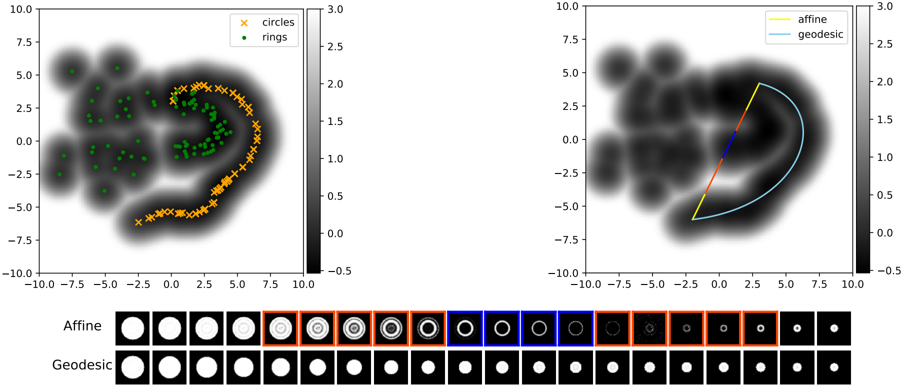
Figure: Riemannian Hamiltonian VAE.

The Learned Latent Space examples

Training samples:

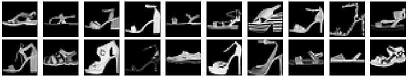


Latent space and interpolations:

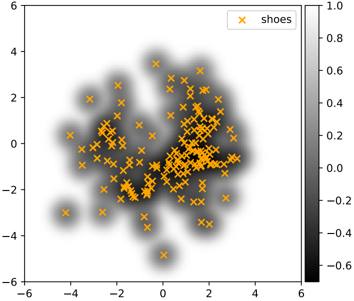


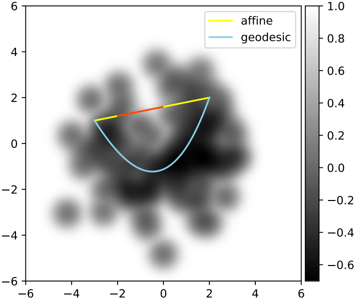
The Learned Latent Space examples

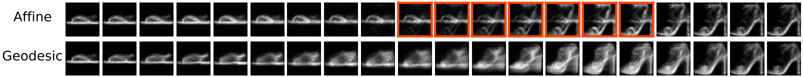
Training samples:



Latent space and interpolations:







3) Improve Data Generation - Sample With the Metric

Idea:

- Use a geometry-based sampling procedure: pdf driven by the metric

$$p(z) = \frac{\mathbb{1}_S(z)\sqrt{\det \mathbf{G}^{-1}(z)}}{\int\limits_{\mathbb{R}^d} \mathbb{1}_S(z)\sqrt{\det \mathbf{G}^{-1}(z) dz}} ,$$

  where S is a compact set and $\mathbb{1}_S(z) = 1$ if $z \in S$, 0 otherwise.
- Use of classic MCMC sampler (e.g. Hamiltonian Monte Carlo)

3) Improve Data Generation - Sample With the Metric

Idea:

- Use a geometry-based sampling procedure: pdf driven by the metric

$$p(z) = \frac{\mathbb{1}_S(z)\sqrt{\det \mathbf{G}^{-1}(z)}}{\int\limits_{\mathbb{R}^d} \mathbb{1}_S(z)\sqrt{\det \mathbf{G}^{-1}(z) dz}} ,$$

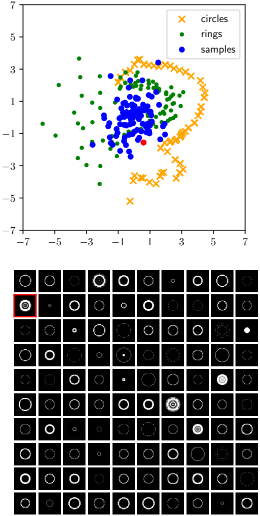
  where S is a compact set and $\mathbb{1}_S(z) = 1$ if $z \in S$, 0 otherwise.
- Use of classic MCMC sampler (e.g. Hamiltonian Monte Carlo)

Pros:

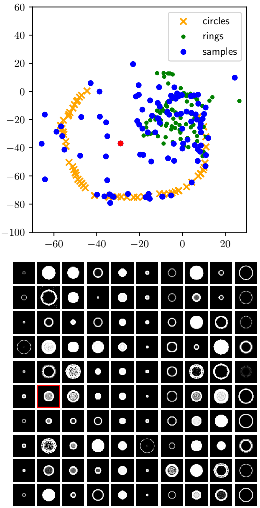
- $\mathbf{G}^{-1}$ easily computable
- Samples "close" to the data

Sampling Comparison

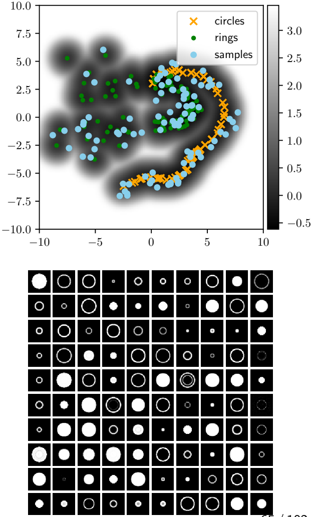
(a) VAE - $\mathcal{N}(0, I)$



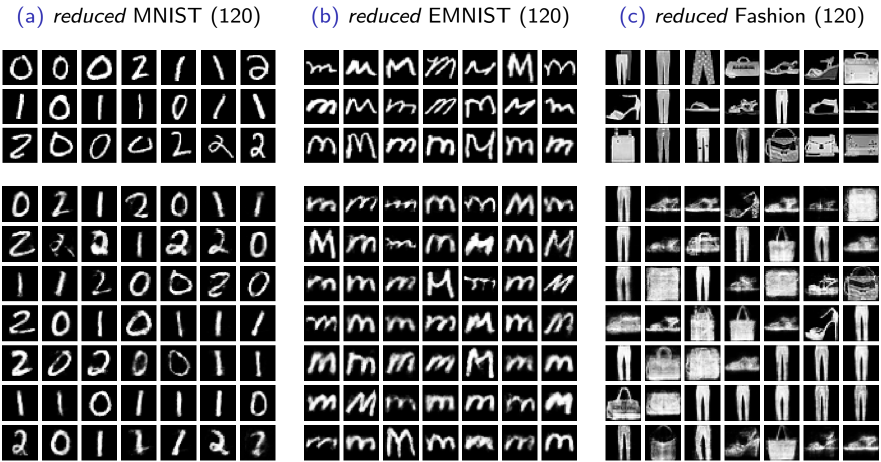
(b) VAE - VAMP (multimodal conditional prior)



(c) Ours



Sampling Comparison - Higher Dimension



(a) *reduced* MNIST (120)

(b) *reduced* EMNIST (120)

(c) *reduced* Fashion (120)

Data Augmentation

1. Framework
2. Toy Data
3. Medical Imaging

Data Augmentation - Framework

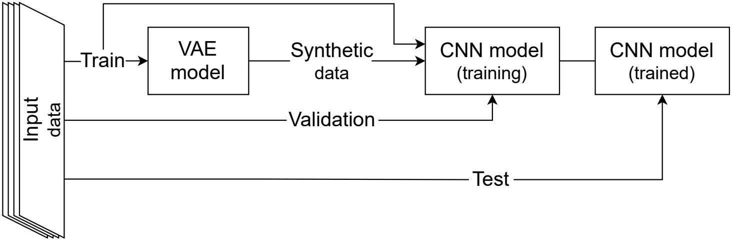


Figure: Data Augmentation pipeline

Performances are estimated using cross-validation.

Data Augmentation

1. Framework
2. Toy Data
3. Medical Imaging

Robustness Across Data Sets

Table: Classification results on *reduced* data sets ($\sim$ 50 samples per class)

	MNIST	MNIST (unbal.)	EMNIST (unbal.)	FASHION
Baseline	89.9 ± 0.6	81.5 ± 0.7	82.6 ± 1.4	76.0 ± 1.5
Baseline + Synthetic				
Basic Augmentation (X5)	92.8 ± 0.4	86.5 ± 0.9	85.6 ± 1.3	77.5 ± 2.0
Basic Augmentation (X10)	88.2 ± 2.2	82.0 ± 2.4	85.7 ± 0.3	79.2 ± 0.6
Basic Augmentation (X15)	92.8 ± 0.7	85.8 ± 3.4	86.6 ± 0.8	80.0 ± 0.5
VAE - 200*	88.5 ± 0.9	84.0 ± 2.0	81.7 ± 3.0	78.6 ± 0.4
VAE - 2k*	92.2 ± 1.6	88.0 ± 2.2	86.0 ± 0.2	79.3 ± 1.1
Ours-200	91.0 ± 1.0	84.1 ± 2.0	85.1 ± 1.1	77.0 ± 0.8
Ours-500	92.3 ± 1.1	87.7 ± 0.9	85.1 ± 1.1	78.5 ± 0.9
Ours-1k	93.2 ± 0.8	**89.7 ± 0.8**	87.0 ± 1.0	**80.2 ± 0.8**
Ours-2k	**94.3 ± 0.8**	89.1 ± 1.9	**87.6 ± 0.8**	78.1 ± 1.8

* Using a standard normal prior to generate

- Classic DA is data set dependent
- Vanilla VAE performs as well as classic DA

Robustness Across Data Sets

Table: Classification results on *reduced* data sets ($\sim$ 50 samples per class) **on synthetic samples only**

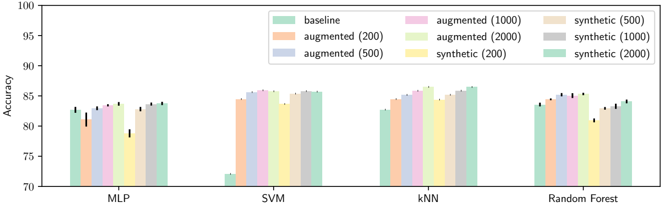
	MNIST	MNIST (unbal.)	EMNIST (unbal.)	FASHION
Baseline	89.9 ± 0.6	81.5 ± 0.7	82.6 ± 1.4	76.0 ± 1.5
Synthetic Only				
VAE - 200*	69.9 ± 1.5	64.6 ± 1.8	65.7 ± 2.6	73.9 ± 3.0
VAE - 2k*	86.5 ± 2.2	79.6 ± 3.8	78.8 ± 3.0	76.7 ± 1.6
Ours-200	87.2 ± 1.1	79.5 ± 1.6	77.0 ± 1.6	77.0 ± 0.8
Ours-500	89.1 ± 1.3	80.4 ± 2.1	80.2 ± 2.0	78.5 ± 0.8
Ours-1k	90.1 ± 1.4	86.2 ± 1.8	82.6 ± 1.3	79.3 ± 0.6
Ours-2k	92.6 ± 1.1	87.5 ± 1.3	86.0 ± 1.0	78.3 ± 0.9

* Using a standard normal prior to generate

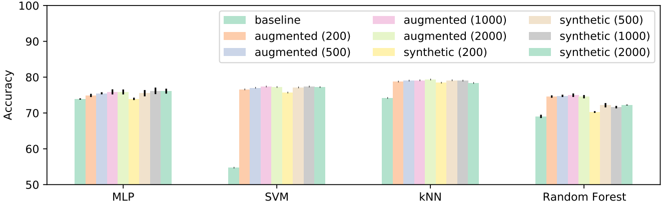
- The proposed model seems to create diverse samples relevant to the classifier

Robustness Across Classifiers

(a) *reduced* MNIST balanced



(b) *reduced* MNIST unbalanced



A Note on the Method Scalability

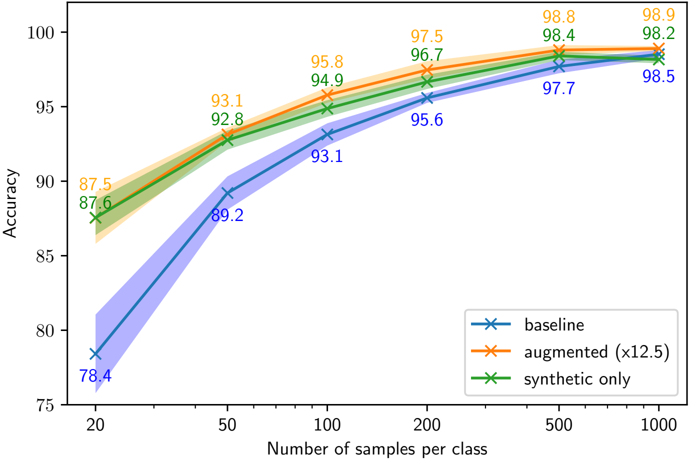


Figure: Benchmark classifier accuracy according to the number of samples in the training set on MNIST.

Data Augmentation

1. Framework

2. Toy Data

3. Medical Imaging

Datasets and classification task

Classification task: Alzheimer's disease patients (**AD**) vs Cognitively Normal participants (**CN**) using T1-weighted MR images.

Table: Summary of participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline.

Data set	Label	Obs.	Age	Sex M/F	MMSE	CDR
ADNI	CN	403	73.3 ± 6.0	185/218	29.1 ± 1.1	0: 403
	AD	362	74.9 ± 7.9	202/160	23.1 ± 2.1	0.5: 169, 1: 192, 2: 1
AIBL	CN	429	73.0 ± 6.2	183/246	28.8 ± 1.2	0: 406, 0.5: 22, 1: 1
	AD	76	74.4 ± 8.0	33/43	20.6 ± 5.5	0.5: 31, 1: 36, 2: 7, 3: 2

MRI preprocessing

Bias field correction (N4ITK) + linear registration (ANTS) + cropping

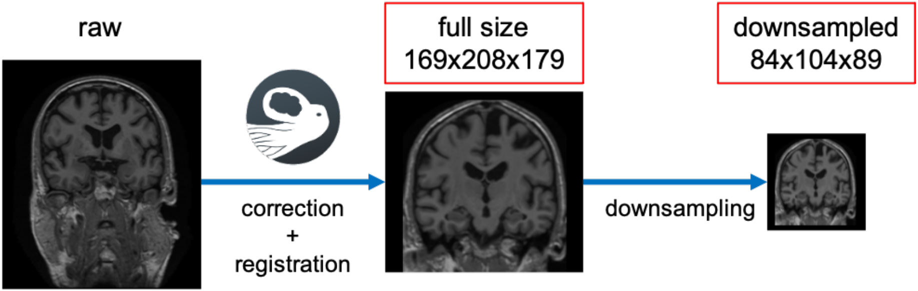


Figure: Preprocessed MRI used in the study

Find wonderful data at:
`/network/lustre/dtlake01/aramis/datasets/adni/caps/caps_v2021`

Synthesized images

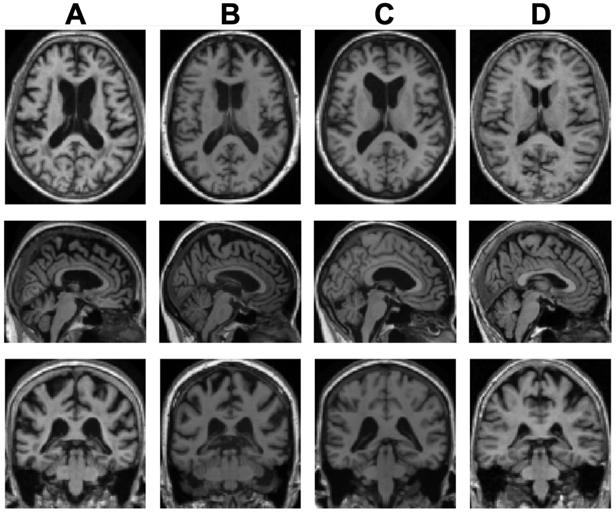


Figure: Example of two *true* patients compared to two generated by our method. Can you find the intruders ?

Synthesized images

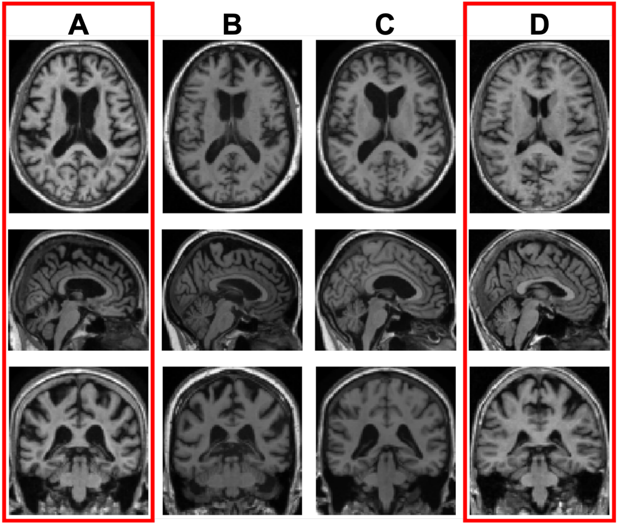
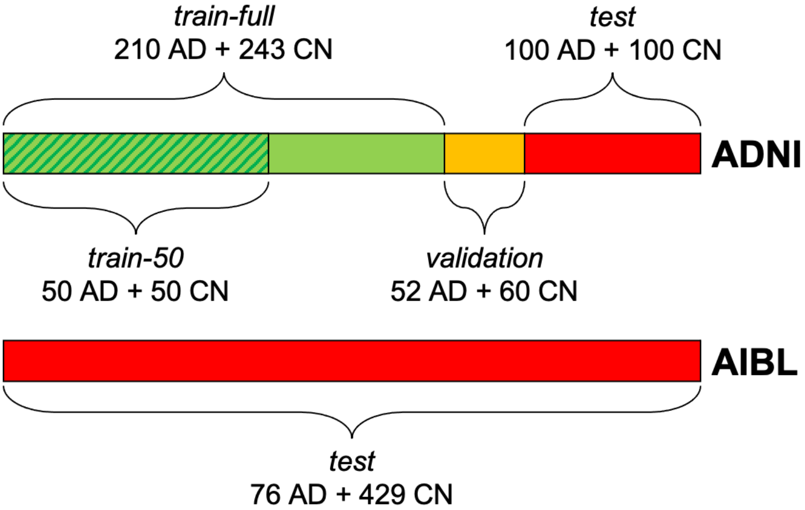


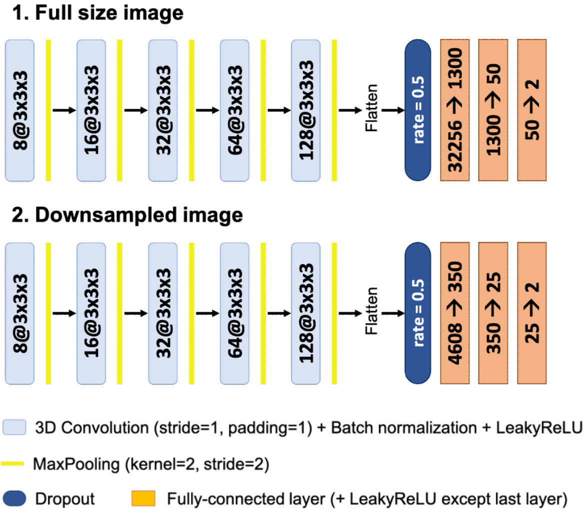
Figure: Example of two *true* patients compared to two generated by our method. Can you find the intruders ?

Evaluation procedure



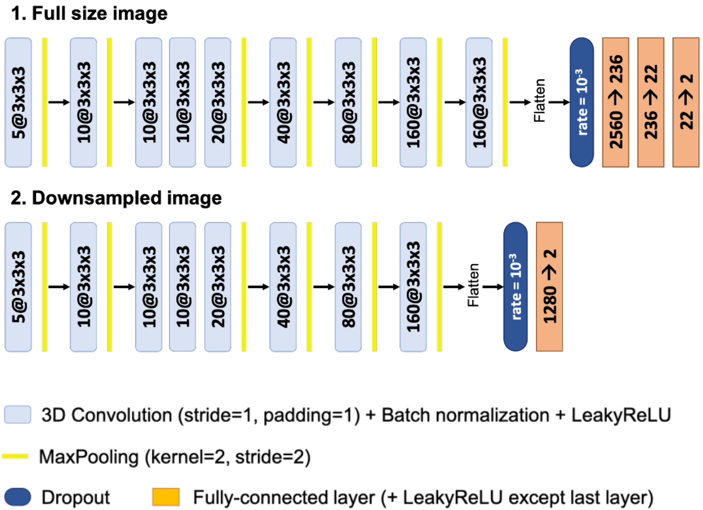
CNN architectures for classifier

Baseline architectures provided by a previous study [WTSDM+20]



CNN architectures for classifier

Optimized architectures opitmize with random search procedure for this training set (ClinicaDL)



1. Full size image

5@3x3x3 → 10@3x3x3 → 10@3x3x3 10@3x3x3 20@3x3x3 → 40@3x3x3 → 80@3x3x3 → 160@3x3x3 → 160@3x3x3 → Flatten → rate = 10^{-3} 2560 → 236 236 → 22 22 → 2

2. Downsampled image

5@3x3x3 → 10@3x3x3 → 10@3x3x3 10@3x3x3 20@3x3x3 → 40@3x3x3 → 80@3x3x3 → 160@3x3x3 → Flatten → rate = 10^{-3} 1280 → 2

3D Convolution (stride=1, padding=1) + Batch normalization + LeakyReLU

MaxPooling (kernel=2, stride=2)

Dropout

Fully-connected layer (+ LeakyReLU except last layer)

Experiments

Four series of experiments:

- **baseline** architecture on *train-50*
- **baseline** architecture on *train-full*
- **optimized** architecture on *train-50*
- **optimized** architecture on *train-full*

For each experiment 20 CNNs are run and the performance is the mean value of the 20 performance values.

Results on train-50 with baseline CNN

Table: Mean test performance of each series of 20 runs trained with the **baseline** hyperparameters on *train-50* set.

data set	ADNI balanced accuracy	AIBL balanced accuracy
real	66.3 ± 2.4	67.2 ± 4.1
real (high-resolution)	67.9 ± 2.3	66.5 ± 3.0
500 synthetic + real	69.4 ± 1.6	68.5 ± 2.5
1000 synthetic + real	70.5 ± 2.1	70.6 ± 3.1
2000 synthetic + real	71.2 ± 1.6	72.8 ± 2.2
3000 synthetic + real	72.6 ± 1.6	73.6 ± 3.0
5000 synthetic + real	**74.1 ± 2.2**	**76.1± 3.6**
10000 synthetic + real	74.0 ± 2.7	74.9 ± 3.2

Increase of balanced accuracy of 6.2 points on ADNI and 8.9 points on AIBL

Results on train-full with baseline CNN

Table: Mean test performance of each series of 20 runs trained with the **baseline** hyperparameters on *train-full* set.

data set	ADNI balanced accuracy	AIBL balanced accuracy
real	77.7 ± 2.5	78.4 ± 2.4
real (high-resolution)	80.6 ± 1.1	80.4 ± 2.6
500 synthetic + real	82.2 ± 2.4	82.9 ± 2.5
1000 synthetic + real	84.4 ± 1.8	83.7 ± 2.3
2000 synthetic + real	85.9 ± 1.6	83.8 ± 2.2
3000 synthetic + real	85.8 ± 1.7	84.4 ± 1.8
5000 synthetic + real	85.7 ± 2.1	84.2 ± 2.2
10000 synthetic + real	**86.3± 1.8**	**85.1± 1.9**

Increase of balanced accuracy of 5.7 points on ADNI and 4.7 on AIBL

Results on train-50 with optimized CNN

Table: Mean test performance of each series of 20 runs trained with the **optimized** hyperparameters on *train-50* set.

data set	ADNI balanced accuracy	AIBL balanced accuracy
real	75.5 ± 2.7	75.6 ± 4.1
real (high-resolution)	72.1 ± 3.1	71.2 ± 5.1
500 synthetic + real	75.6 ± 2.5	76.0 ± 4.2
1000 synthetic + real	77.8 ± 2.3	80.9 ± 3.2
2000 synthetic + real	76.9 ± 2.4	80.0 ± 3.6
3000 synthetic + real	77.8 ± 1.9	81.2 ± 3.7
5000 synthetic + real	76.9 ± 2.5	80.9 ± 2.7
10000 synthetic + real	**78.0±2.1**	**81.9±2.2**

Increase of balanced accuracy of 2.5 points on ADNI and 6.3 points on AIBL

Results on train-full with optimized CNN

Table: Mean test performance of each series of 20 runs trained with the **optimized** hyperparameters on *train-full* set.

data set	ADNI balanced accuracy	AIBL balanced accuracy
real	85.5 ± 2.4	81.9 ± 3.2
real (high-resolution)	85.7 ± 2.5	84.4 ± 1.7
500 synthetic + real	86.0 ± 1.8	83.2 ± 2.4
1000 synthetic + real	86.5 ± 1.9	83.7 ± 2.0
2000 synthetic + real	**87.2±1.7**	84.0 ± 2.0
3000 synthetic + real	85.8 ± 2.6	83.6 ± 3.2
5000 synthetic + real	86.4 ± 1.3	83.5 ± 2.2
10000 synthetic + real	86.7 ± 1.8	**84.3±1.8**

Increase of balanced accuracy of 1.5 point on ADNI and -0.1 point on AIBL

Conclusion

We have proposed

- a new geometry aware VAE-based data augmentation framework relevant for representing and classifying data in the HDLSS setting.
- Validated on classification tasks on *toy* and *real-life* data sets in particular in the *High dimension low sample size setting*.

Conclusion

We have proposed

- a new geometry aware VAE-based data augmentation framework relevant for representing and classifying data in the HDLSS setting.
- Validated on classification tasks on *toy* and *real-life* data sets in particular in the *High dimension low sample size setting*.

Strengths:

- Independent on the nature of the data set: from 2D images (MNIST, EMNIST, FASHION) to 3D medical images (ADNI and AIBL),
- Relevant synthetic data: classifiers achieved a similar or better classification performance when trained only on synthetic data than on the *real* train set.
- Classifier independence: MLP, random forest, k-NN and SVM (on toy data sets) ; baseline and optimized parameters (on medical images).

Conclusion

We have proposed

- a new geometry aware VAE-based data augmentation framework relevant for representing and classifying data in the HDLSS setting.
- Validated on classification tasks on *toy* and *real-life* data sets in particular in the *High dimension low sample size setting*.

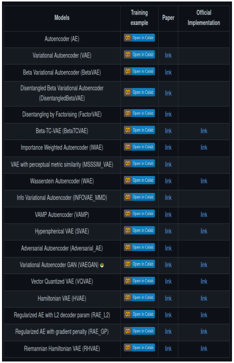
Limitations - what could be improved:

- No extensive search on VAE architecture.
- Would it benefit from the use of longitudinal data?
- *train-50* is still large compared to some medical data sets. . .

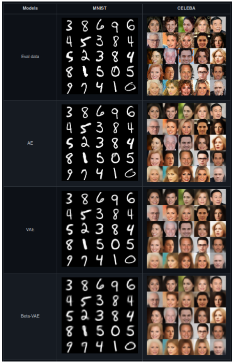
Implementation available https://clementchadebec.github.io/projects/

PyRaug

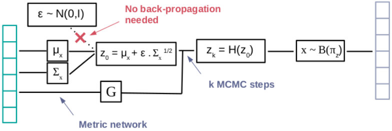
AND Extensive comparison of data generation based on VAEs



Models	Training example	Paper	Official Implementation
Autoencoder (AE)	Open in Colab		
Variational Autoencoder (VAE)	Open in Colab	link	
Beta Variational Autoencoder (BetaVAE)	Open in Colab	link	
Disentangled Beta Variational Autoencoder (DisentangledBetaVAE)	Open in Colab	link	
Disentangling by Factorising (FactorVAE)	Open in Colab	link	
Beta-TC-VAE (BetaTCVAE)	Open in Colab	link	link
Importance Weighted Autoencoder (IWAE)	Open in Colab	link	link
VAE with perceptual metric similarity (MSSSIM_VAE)	Open in Colab	link	
Wasserstein Autoencoder (WAE)	Open in Colab	link	link
Info Variational Autoencoder (INFOVAE_MMD)	Open in Colab	link	
VAMP Autoencoder (VAMP)	Open in Colab	link	link
Hyperspherical VAE (SVAE)	Open in Colab	link	link
Adversarial Autoencoder (Adversarial_AE)	Open in Colab	link	
Variational Autoencoder GAN (VAEGAN)	Open in Colab	link	link
Vector Quantized VAE (VQVAE)	Open in Colab	link	link
Hamiltonian VAE (HVAE)	Open in Colab	link	link
Regularized AE with L2 decoder param (RAE_L2)	Open in Colab	link	link
Regularized AE with gradient penalty (RAE_GP)	Open in Colab	link	link
Riemannian Hamiltonian VAE (RHVAE)	Open in Colab	link	link



Thank you!



$\varepsilon \sim \mathcal{N}(0,I)$

No back-propagation needed

μ_x

Σ_x

$z_0 = \mu_x + \varepsilon \, . \, \Sigma_x^{1/2}$

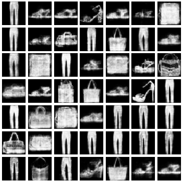
$z_k = H(z_0)$

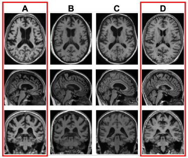
$x \sim \mathcal{B}(\pi_z)$

G

k MCMC steps

Metric network





https://clementchadebec.github.io/projects/

Contacts:

clement.chadebec@inria.fr
stephanie.allassonniere@inria.fr

References I

- Christoph Baur, Shadi Albarqouni, and Nassir Navab, *Generating highly realistic images of skin lesions with GANs*, OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis, Springer, 2018, pp. 260–267.

- Lei Bi, Jinman Kim, Ashnil Kumar, Dagan Feng, and Michael Fulham, *Synthesis of Positron Emission Tomography (PET) Images via Multi-channel Generative Adversarial Networks (GANs)*, Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment, LNCS, Springer, 2017, pp. 43–51.

- Anthony L Caterini, Arnaud Doucet, and Dino Sejdinovic, *Hamiltonian variational auto-encoder*, Advances in Neural Information Processing Systems, 2018, pp. 8167–8177.

References II

- Francesco Calimeri, Aldo Marzullo, Claudio Stamile, and Giorgio Terracina, *Biomedical data augmentation using generative adversarial neural networks*, International conference on artificial neural networks, Springer, 2017, pp. 626–634.

- Clément Chadebec, Elina Thibeau-Sutre, Ninon Burgos, and Stéphanie Allassonnière, *Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder*, arXiv preprint arXiv:2105.00026 (2021).

- Alain Durmus, Eric Moulines, and Eero Saksman, *On the convergence of hamiltonian monte carlo*, arXiv preprint arXiv:1705.00166 (2017).

- Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan, *GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification*, Neurocomputing **321** (2018), 321–331.

References III

- Mark Girolami and Ben Calderhead, *Riemann manifold langevin and hamiltonian monte carlo methods*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73** (2011), no. 2, 123–214.
- Mark Girolami, Ben Calderhead, and Siu A Chin, *Riemannian manifold hamiltonian monte carlo*, arXiv preprint arXiv:0907.1100 (2009).
- Dimitrios Korkinof, Tobias Rijken, Michael O'Neill, Joseph Yearsley, Hugh Harvey, and Ben Glocker, *High-resolution mammogram synthesis using progressive generative adversarial networks*, arXiv preprint arXiv:1807.03401 (2018).
- Diederik P. Kingma and Max Welling, *Auto-encoding variational bayes*, arXiv:1312.6114 [cs, stat] (2014).
- Samuel Livingstone, Michael Betancourt, Simon Byrne, Mark Girolami, and others, *On the geometric ergodicity of hamiltonian monte carlo*, Bernoulli **25** (2019), no. 4, 3109–3138.

References IV

- Maxime Louis, *Computational and statistical methods for trajectory analysis in a Riemannian geometry setting*, PhD Thesis, Sorbonnes universités, 2019.
- Yufei Liu, Yuan Zhou, Xin Liu, Fang Dong, Chang Wang, and Zihong Wang, *Wasserstein gan-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology*, Engineering **5** (2019), no. 1, 156–163.
- Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood, *Chest x-ray generation and data augmentation for cardiovascular abnormality classification*, Medical Imaging 2018: Image Processing, vol. 10574, International Society for Optics and Photonics, 2018, p. 105741M.
- Radford M Neal and others, *MCMC using hamiltonian dynamics*, Handbook of Markov Chain Monte Carlo **2** (2011), no. 11, 2.

References V

- Danilo Rezende and Shakir Mohamed, *Variational inference with normalizing flows*, International Conference on Machine Learning, PMLR, 2015, pp. 1530–1538.

- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra, *Stochastic backpropagation and approximate inference in deep generative models*, International conference on machine learning, PMLR, 2014, pp. 1278–1286.

- Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski, *Medical image synthesis for data augmentation and anonymization using generative adversarial networks*, International Workshop on Simulation and Synthesis in Medical Imaging, LNCS, Springer, 2018, pp. 1–11.

References VI

- Hojjat Salehinejad, Shahrokh Valaee, Tim Dowdell, Errol Colak, and Joseph Barfett, *Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks*, 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 990–994.

- Veit Sandfort, Ke Yan, Perry J. Pickhardt, and Ronald M. Summers, *Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks*, Scientific reports **9** (2019), no. 1, 16884.

- Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Fadi Al-Turjman, and Plácido Rogerio Pinheiro, *Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection*, Ieee Access **8** (2020), 91916–91923.

References VII

- Junhao Wen, Elina Thibeau-Sutre, Mauricio Diaz-Melo, Jorge Samper-González, Alexandre Routier, Simona Bottani, Didier Dormont, Stanley Durrleman, Ninon Burgos, and Olivier Colliot, *Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation*, Medical Image Analysis **63** (2020), 101694.

- Eric Wu, Kevin Wu, David Cox, and William Lotter, *Conditional infilling gans for data augmentation in mammogram classification*, Image analysis for moving organ, breast, and thoracic images, Springer, 2018, pp. 98–106.

- Xin Yi, Ekta Walia, and Paul Babyn, *Generative adversarial network in medical imaging: A review*, Medical image analysis **58** (2019), 101552.

Clustering

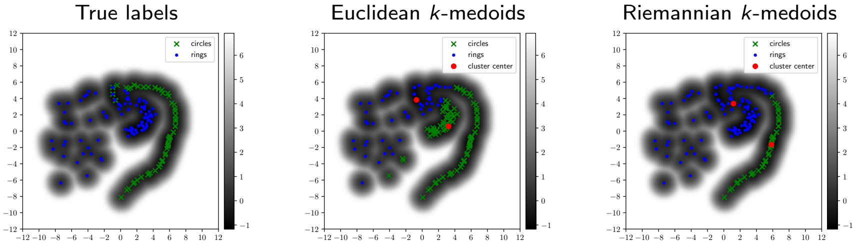


Figure: Euclidean and Riemannian *k*-medoids custering.

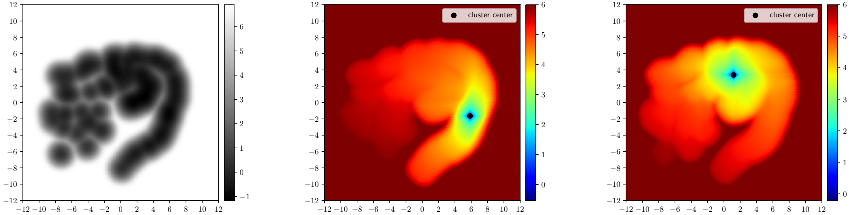


Figure: Distance maps.

Results - Clustering

Data set	Model	Subset 1	Subset 2	Subset 3	Mean
Synthetic data	linear	53.88	62.52	71.63	62.68
	geodesic	**71.41**	**81.39**	**79.49**	**77.43**
MNIST 1	linear	89.73	93.11	91.80	91.55
	geodesic	**91.68**	**94.51**	**95.63**	**93.94**
MNIST 2	linear	68.24	69.22	79.05	71.17
	geodesic	**70.35**	**71.34**	**79.64**	**73.78**
MNIST 3	linear	75.55	75.76	81.70	77.67
	geodesic	**76.08**	**77.94**	**81.96**	**78.66**
FashionMNIST 1	linear	90.47	91.63	86.78	89.63
	geodesic	**91.44**	**92.55**	**87.46**	**90.48**
FashionMNIST 2	linear	92.20	91.26	93.30	92.25
	geodesic	**93.56**	**91.80**	**94.12**	**93.16**
FashionMNIST 3	linear	72.46	79.58	83.16	78.40
	geodesic	**74.89**	**81.88**	**84.83**	**80.53**

Table: F1-Scores.