Learning Manifold-structured Data using Deep networks: Theory and Applications

Rongjie Lai

Rensselaer Polytechnic Institute

Lisbon Webinar Math, Phy & ML

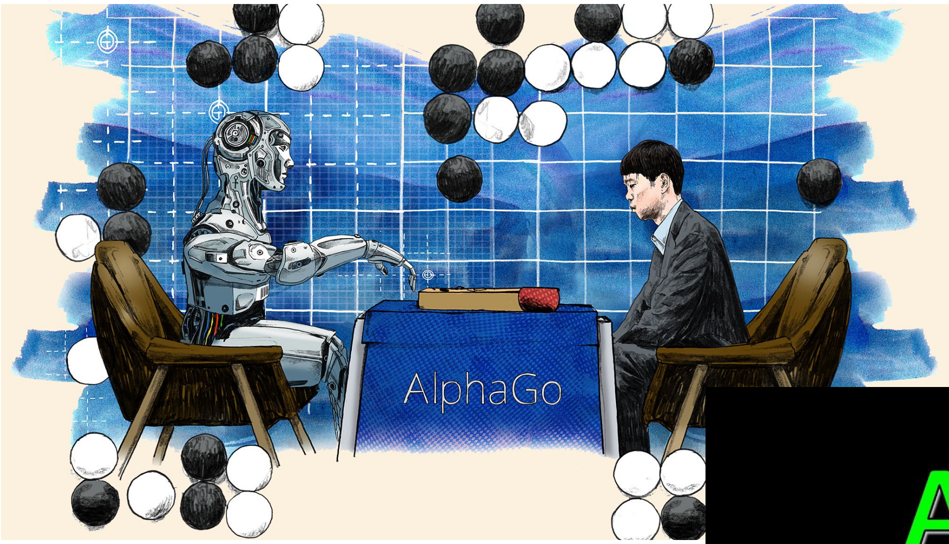
Supported in part by an NSF CAREER Award, NSF SCALE MoDL and IBM AIRC

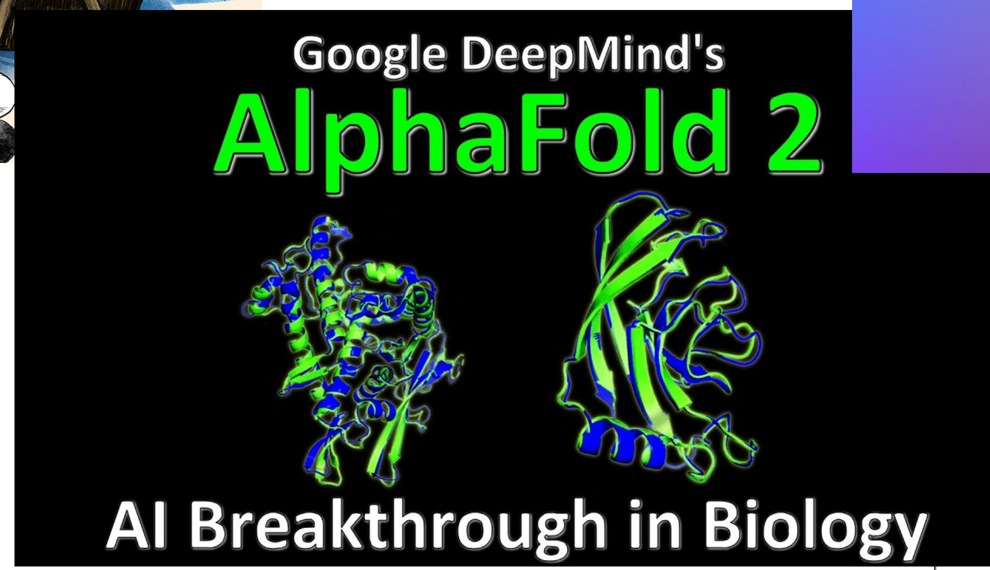
Outline

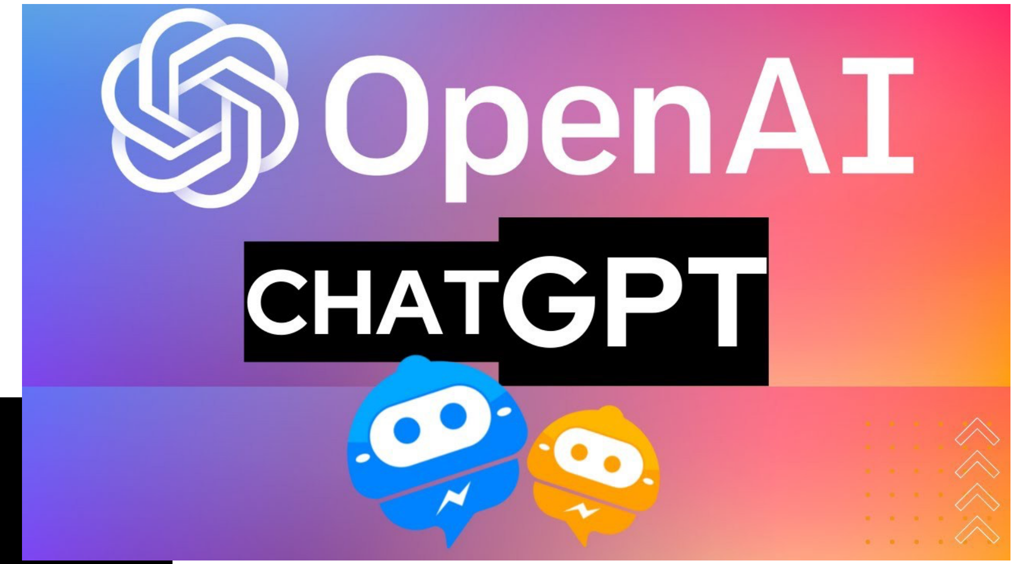
1. **All data points sampled on or near a d-dimensional unknown manifold embedded in R^m. How effective can DNNs learn the manifold structure?**
   (with Schonsheck@RPI, Chen@IBM, A. Hvarilla & W. Liao@Gatech, H. Liu@HKBU)

2. **Each data point is a 2-dimensional manifold: Design spatially convolutional operation on manifolds and conduct deep learning tasks including surface registration, geometric information disentanglement, point clouds classification and segmentation.** (with Schonsheck@RPI, Tatro@RPI, Jin@PKU, Dong@PKU)

Deep Neural Networks







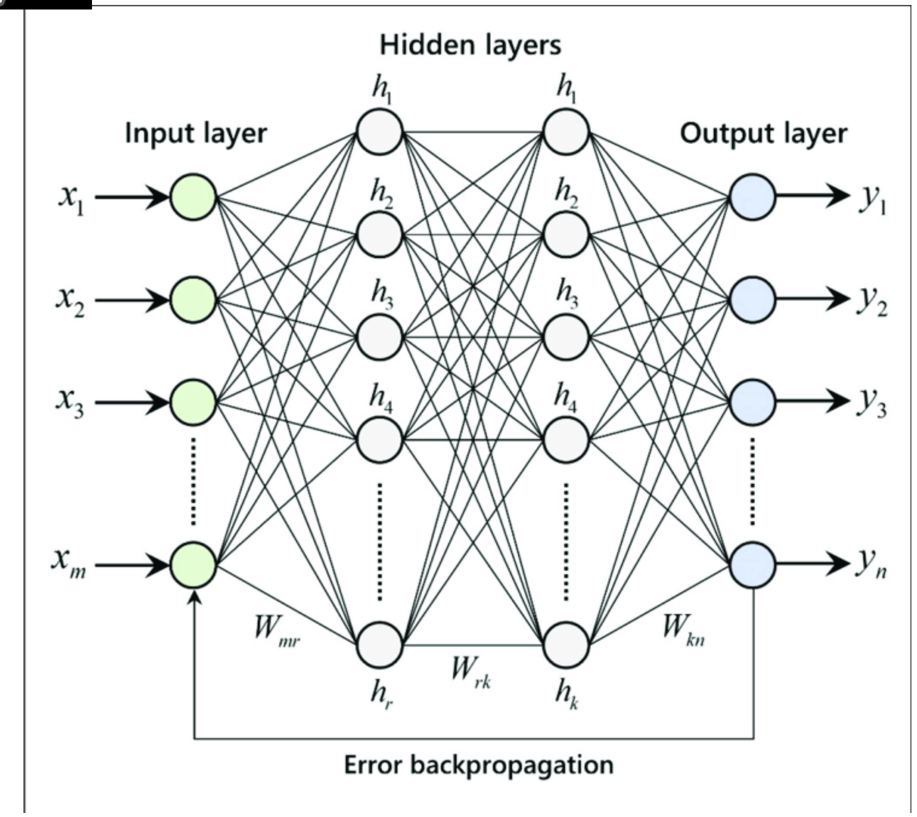
A feedforward network:

$$F_\Theta(x) = f_k \circ \sigma \circ f_{k-1} \cdots \sigma \circ f_1(x)$$

where each $f_i(x) = W_i x + b_i$ and σ nonlinear activation, e.g. $\max\{x, 0\}$

Given $\{(x_i, y_i)\}_{i=1}^n$, Train: $\min_{\Theta=\{W_i, b_i\}} \frac{1}{n} \sum_{k=1}^{n} h(F_\Theta(x_k), y_k)$

For example, h can be squared norm for regression, or cross entropy for classification.



Curse of dimensionality

Deep networks have been very successful in many applications.

Approx. functions: $f : \mathcal{X} \to \mathbb{R}$

Approx. maps (operators): $F : \mathcal{X} \to \mathcal{Y}$

$\mathcal{X}$ is often high dimension, or even a functional space.

- Deep neural networks preform reasonably well. For instance, in the ImageNet challenge, the ambient space dimension m = 224*224*3.

- Consider $\{x_i\}_{i=1}^n$ uniformly sampled in [0, 1]^m. The expected distance to any x

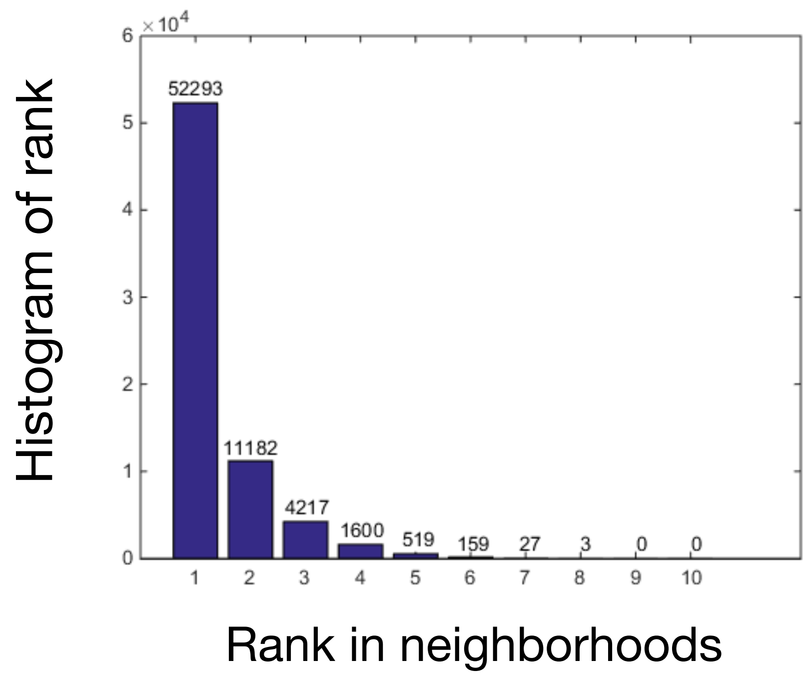
$$\mathbb{E}\{\min_i \|x - x_i\|\} \geq \frac{m}{2(m+1)} \left(\frac{1}{n}\right)^{1/m}$$

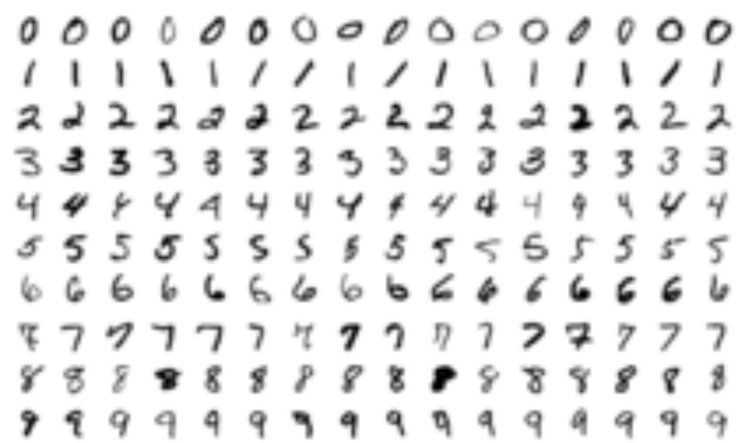
To achieve accuracy ϵ , sample size needs $n \gtrsim 1/\epsilon^m$

	n=100	n=1000	n = 10,000	n=100,000
m=1	2.5 $\times 10^{-3}$	2.5 $\times 10^{-4}$	2.5 $\times 10^{-5}$	2.5 $\times 10^{-6}$
m=20	0.37	0.34	0.30	0.26

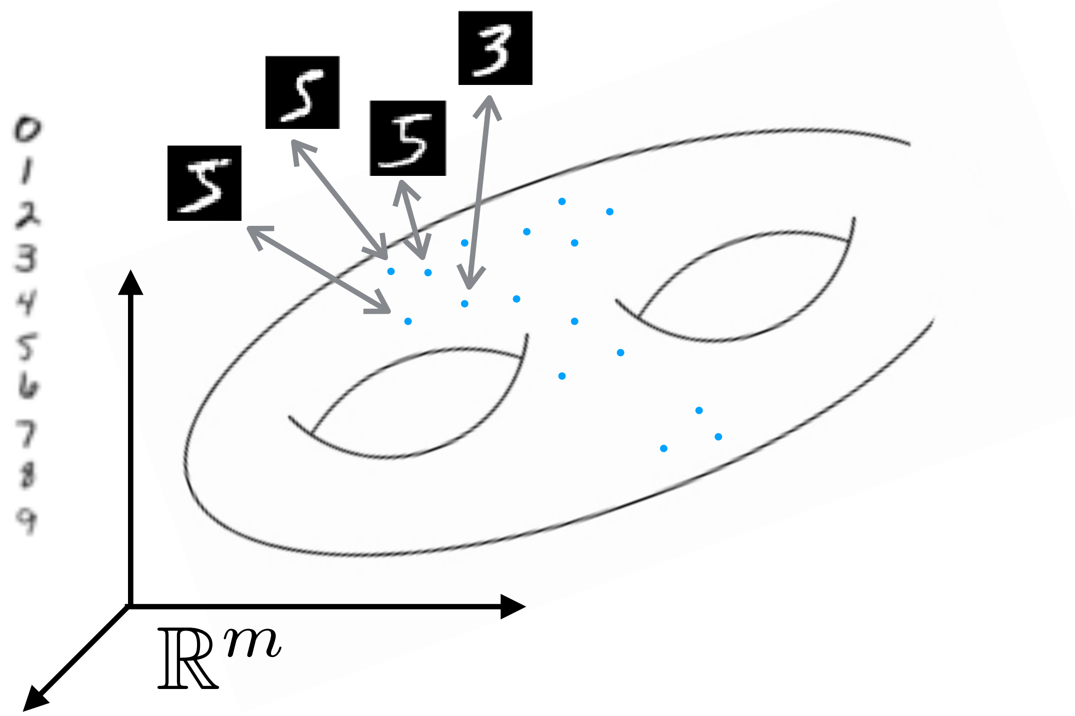
Low dimensional models

Data points sit in a low-dime coherent structure in R^m [Pope et al.]





MNIST



It is commonly believed that DNNs can automatically learn the low-dimension structure

Curse of dimensionality, Coherent data structures → **Low-dimensional Models, $\mathcal{M} \subset \mathbb{R}^n$**

Consider a data set $\{\mathbf{x}_i\}_{i=1}^n$ sampled on or near an unknown d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$. How effective and robust can DNNs learn $\mathcal{M}$?

Some literature

- **Convectional manifold learning methods (not DNN based methods)**

A series of works on manifold learning have been effective on linear dimension reduction of data, including IsoMap (Tenenbaum et al., 2000), Locally Linear Embedding (Roweis and Saul, 2000; Zhang and Wang, 2006), Laplacian Eigenmap (Belkin and Niyogi, 2003), Diffusion map (Coifman et al., 2005), t-SNE (Van der Maaten and Hinton, 2008), Geometric Multi-Resolution Analysis (Allard et al., 2012; Liao and Maggioni, 2019) and many others (Aamari and Levrard, 2019). As extensions, the noisy manifold setting has been studied in (Maggioni et al., 2016; Genovese et al., 2012b,a; Puchkin and Spokoiny, 2022)

- **DNN-based methods. Approximating functions or mapping on $\mathbb{R}^d$ or a known manifold**

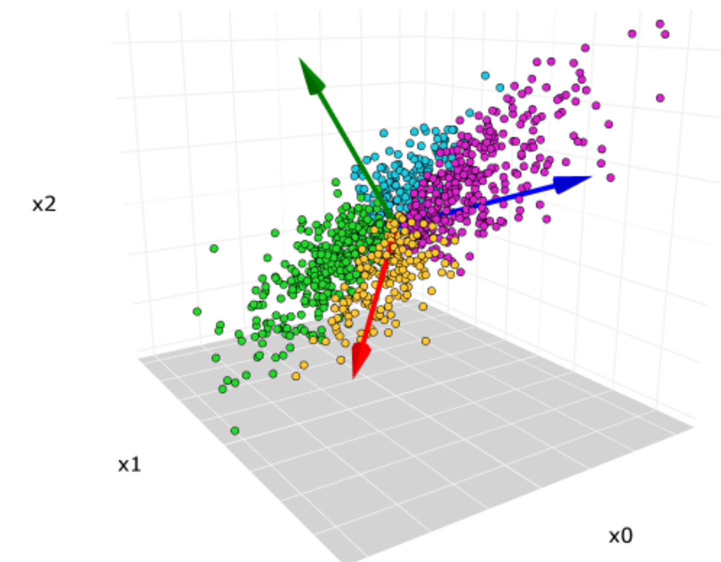
In order to justify the performance of deep neural networks, many mathematical theories have been established on function approximation (Hornik et al., 1989; Yarotsky, 2017; Shaham et al., 2018; Schmidt-Hieber, 2019; Shen et al., 2019; Chen et al., 2019a; Cloninger and Klock, 2021; Montanelli and Yang, 2020; Liu et al., 2022a,c), regression (Chui and Mhaskar, 2018; Chen et al., 2019b; Nakada and Imaizumi, 2020), classification (Liu et al., 2021), operator learning (Liu et al., 2022b) and causal inference on a low-dimensional manifold (Chen et al., 2020).

Dimension reduction and deep generative models

Principle component analysis

Consider a data set $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^m$ sampled from a given distribution ξ. Compute d principle components $\mathbf{u}_1, \cdots, \mathbf{u}_d \in \mathbb{R}^m$. Given $\mathbf{x} \sim \xi$, PCA tells us

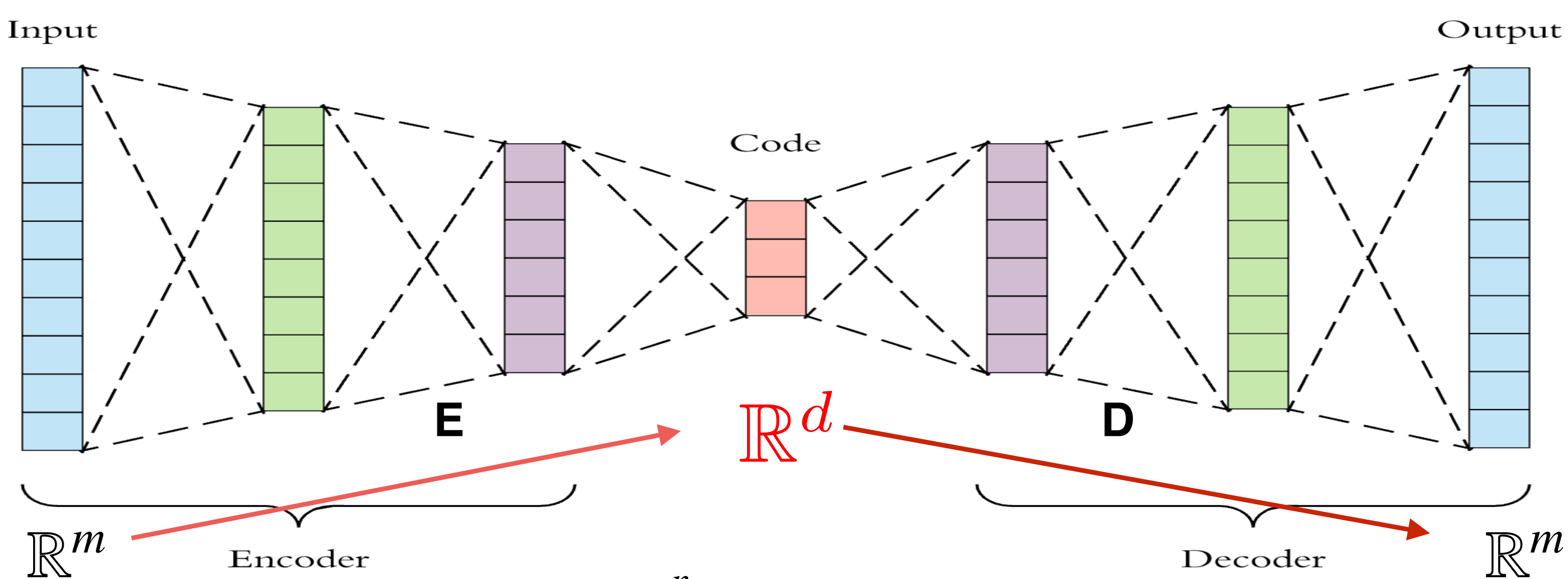
$$\mathbf{x} \approx \sum_{k=1}^{d} \langle \mathbf{x}, \mathbf{u}_k \rangle \mathbf{u}_k$$

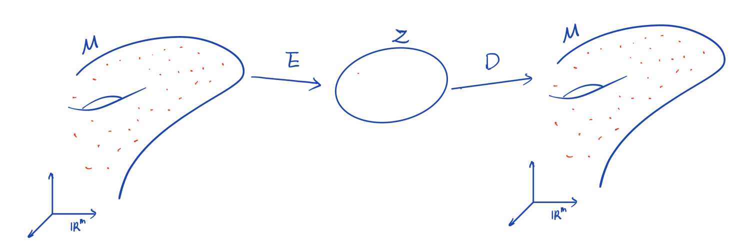


Encoding: $\mathbf{E} : \mathbb{R}^m \to \mathbb{R}^d, \qquad \mathbf{E}(\mathbf{x}) = (\langle \mathbf{x}, \mathbf{u}_1 \rangle, \cdots, \langle \mathbf{x}, \mathbf{u}_d \rangle)$

Decoding $\mathbf{D} : \mathbb{R}^d \to \mathbb{R}^m, \qquad \mathbf{D}(z_1, \cdots, z_d) = \sum_{k=1}^{d} z_k \mathbf{u}_k$

Auto-encoders [Bourlard & Kamp'98, Hinton & Zemel '94, Liou et al'14], Variational auto-encoders [Kingma & Welling'13]

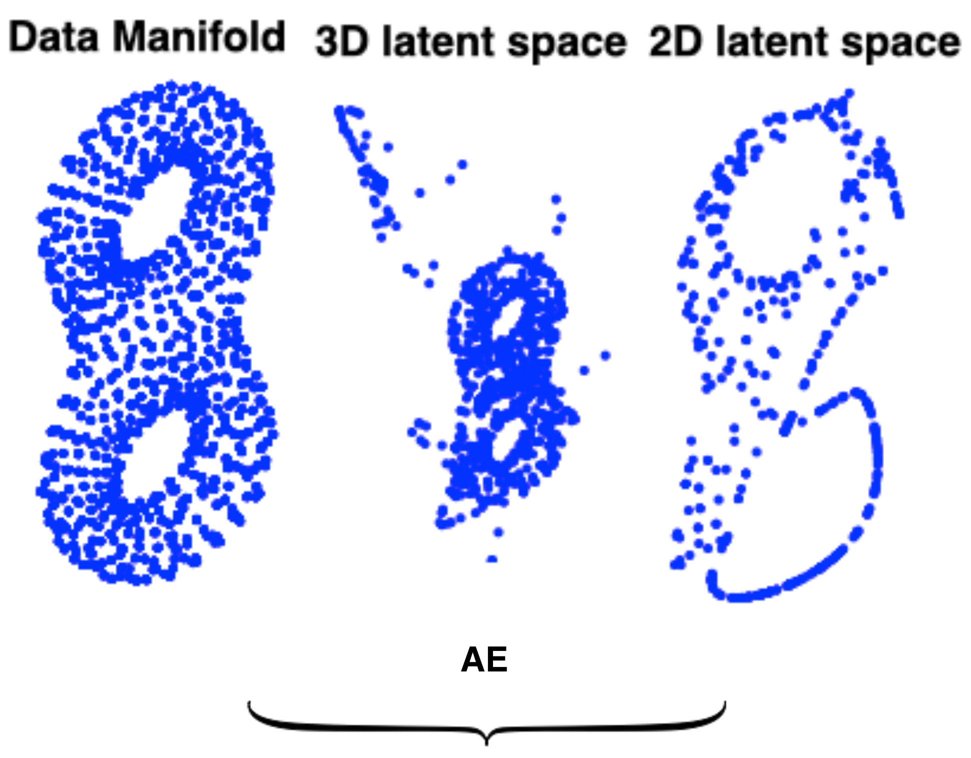


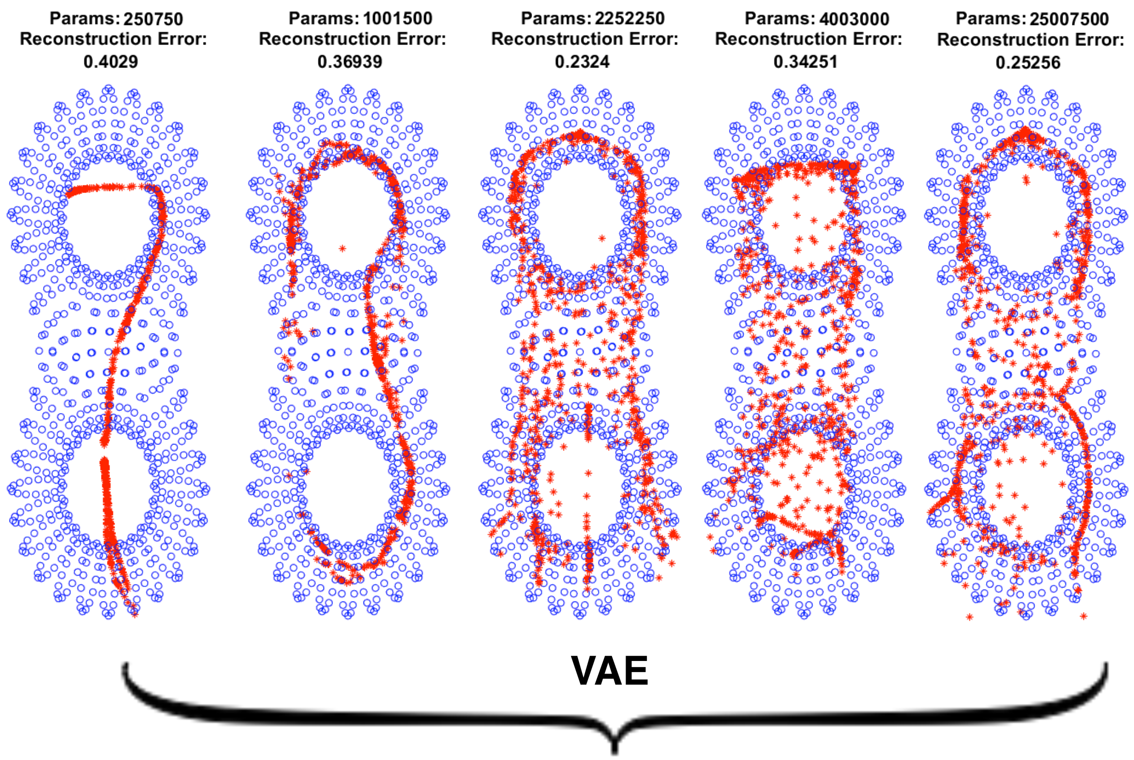


$$\min_{E,D} \frac{1}{n} \sum_{i=1}^{n} \|x_i - D \circ E(x_i)\|^2$$

Latent space structure v.s. Data manifold structure

Given a data set sampled on a double torus, performance of AE and VAE using a flat latent space





Latent space structure v.s. Data manifold structure

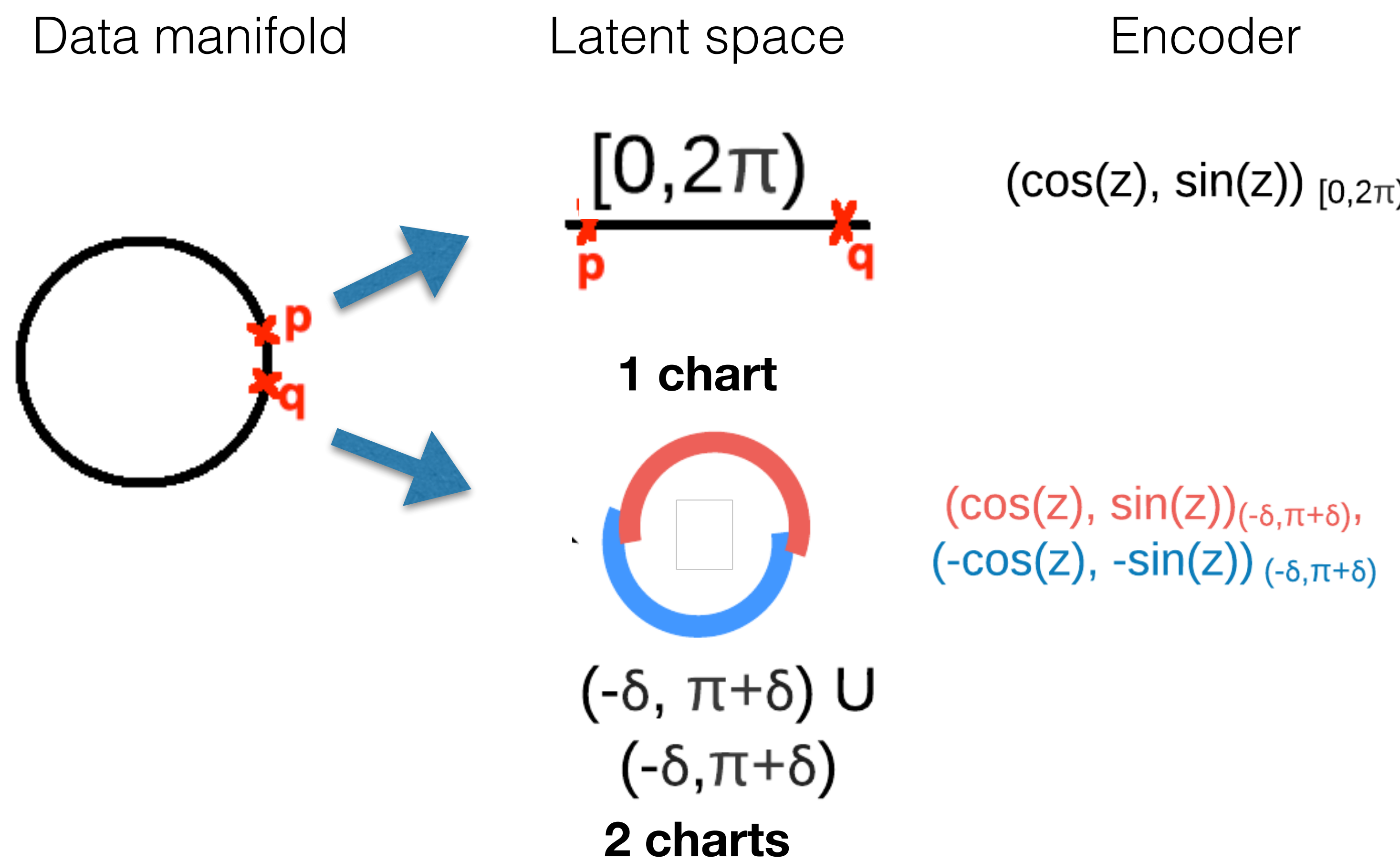
Observations:

1. **A flat domain as the latent space can not cover data manifold well;**

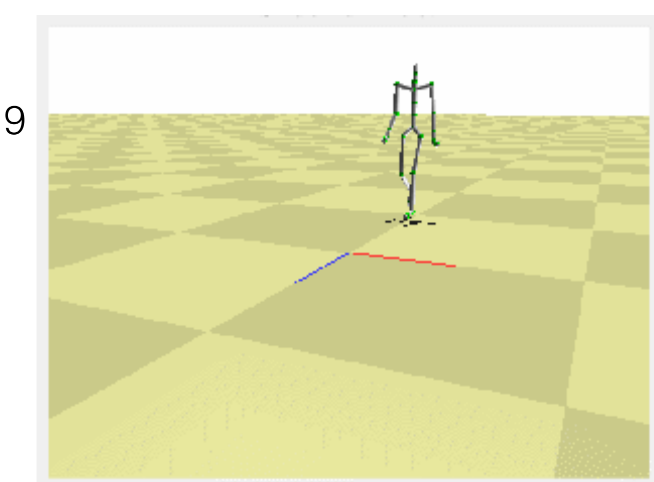
2. **A higher dimension latent space generates undesired data;**

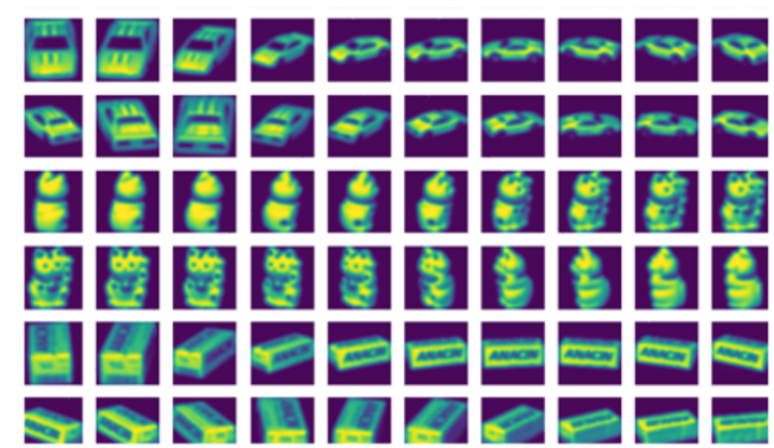
3. **Representation with topology breaking may introduce big metric distortion.**

Structured latent space is needed.



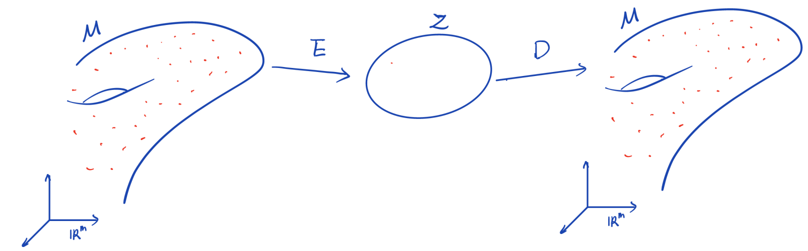
Spherical latent space: Xu-Durrett'18, Davidson et al.'18, Rey et al'19
Closed path: Connor-Rozell'19,
Lie groups (e.g SO(3)): Falorsi et al'18,
Diffusion geometry: Li-Lindenbaum-Cheng-Cloninger'19





Our objectives

Consider $\{x_i\}_{i=1}^n$ sampled on a compact d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$ with possible noise

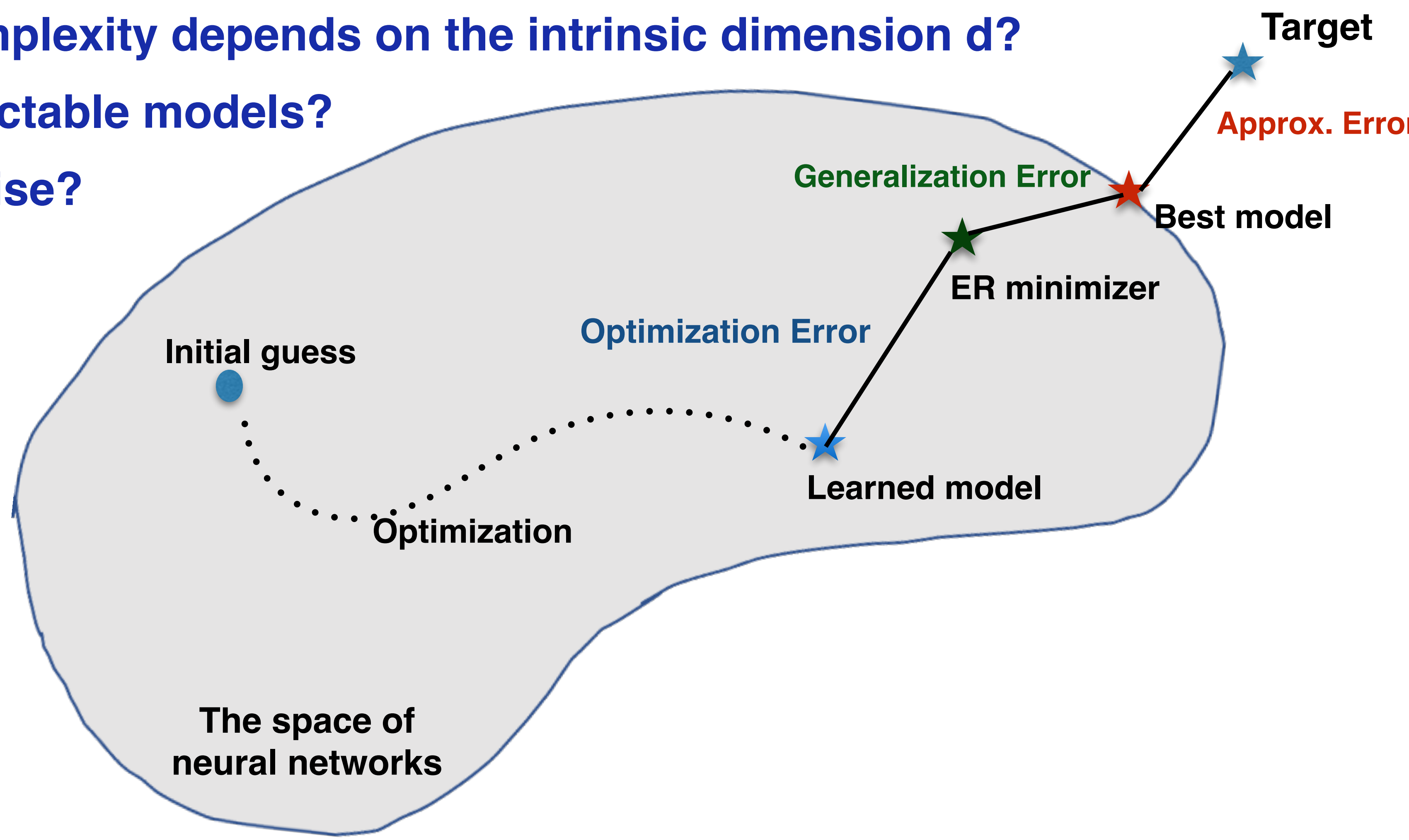


Empirical Risk (ER) minimization: $$(\hat{\mathbf{E}}, \hat{\mathbf{D}}) = \arg\min_{\mathbf{E},\mathbf{D}} \frac{1}{n}\sum_{i=1}^{n} \|x_i - \mathbf{D} \circ \mathbf{E}(x_i)\|^2$$

Approximation Error: The smallest possible error $\|x - \tilde{\mathbf{D}} \circ \tilde{\mathbf{E}}(x)\|$ for all test data $x \in \mathcal{M}$

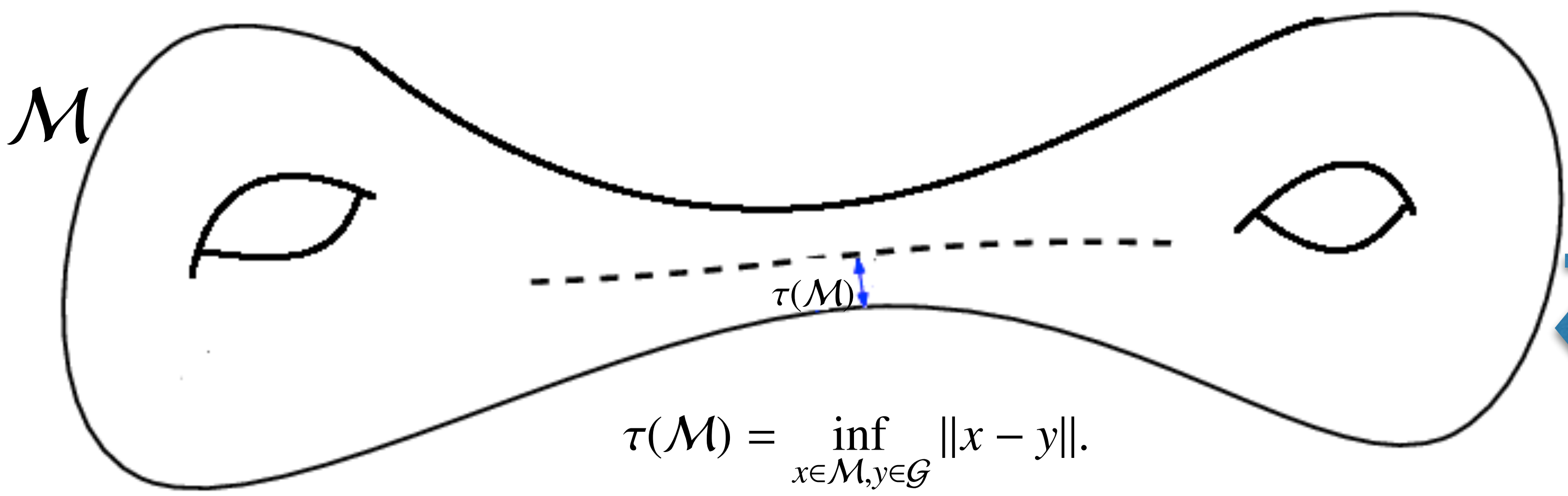
Generalization Error: Given a minimizer $(\hat{\mathbf{E}}, \hat{\mathbf{D}})$ from ER, consider $\|x - \hat{\mathbf{D}} \circ \hat{\mathbf{E}}(x)\|$ for all test data $x \in \mathcal{M}$

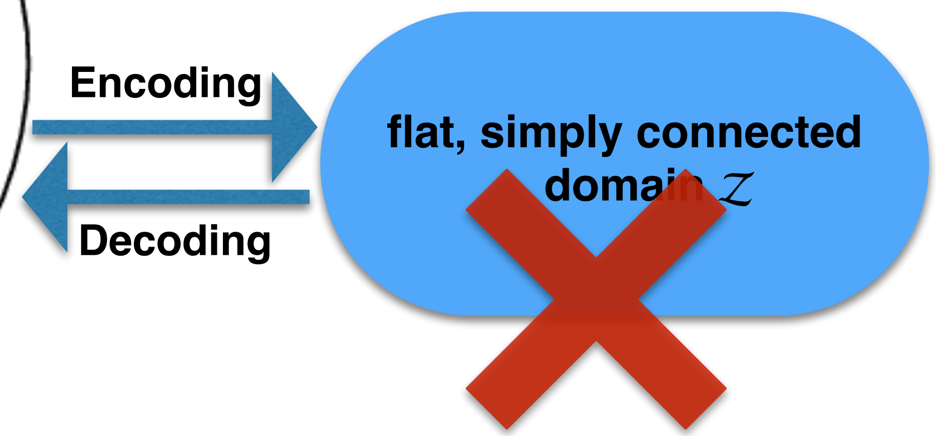
- **Existence of such an network (network structure) for a data manifold?**
- **Error/network complexity depends on the intrinsic dimension d?**
- **Computational tractable models?**
- **Robustness to noise?**



Topology requirement under faithful representation

Definition 1 (Faithful Representation). *An auto-encoder* $(\mathcal{Z}; \boldsymbol{E}, \boldsymbol{D})$ *is called a faithful representation of* $\mathcal{M}$ *if* $x = \boldsymbol{D} \circ \boldsymbol{E}(x), \forall x \in \mathcal{M}$. *An auto-encoder is called an* ϵ*-faithful representation of* $\mathcal{M}$ *if* $\sup_{x \in \mathcal{M}} \|x - \boldsymbol{D} \circ \boldsymbol{E}(x)\| \leq \epsilon$.





Medial axis $\mathcal{G} = \left\{ y \in \mathbb{R}^m \mid \exists p \neq q \in \mathcal{M} \text{ s.t. } \|y - p\| = \|y - q\| = \inf_{x \in \mathcal{M}} \|x - y\| \right\}$

Not necessarily

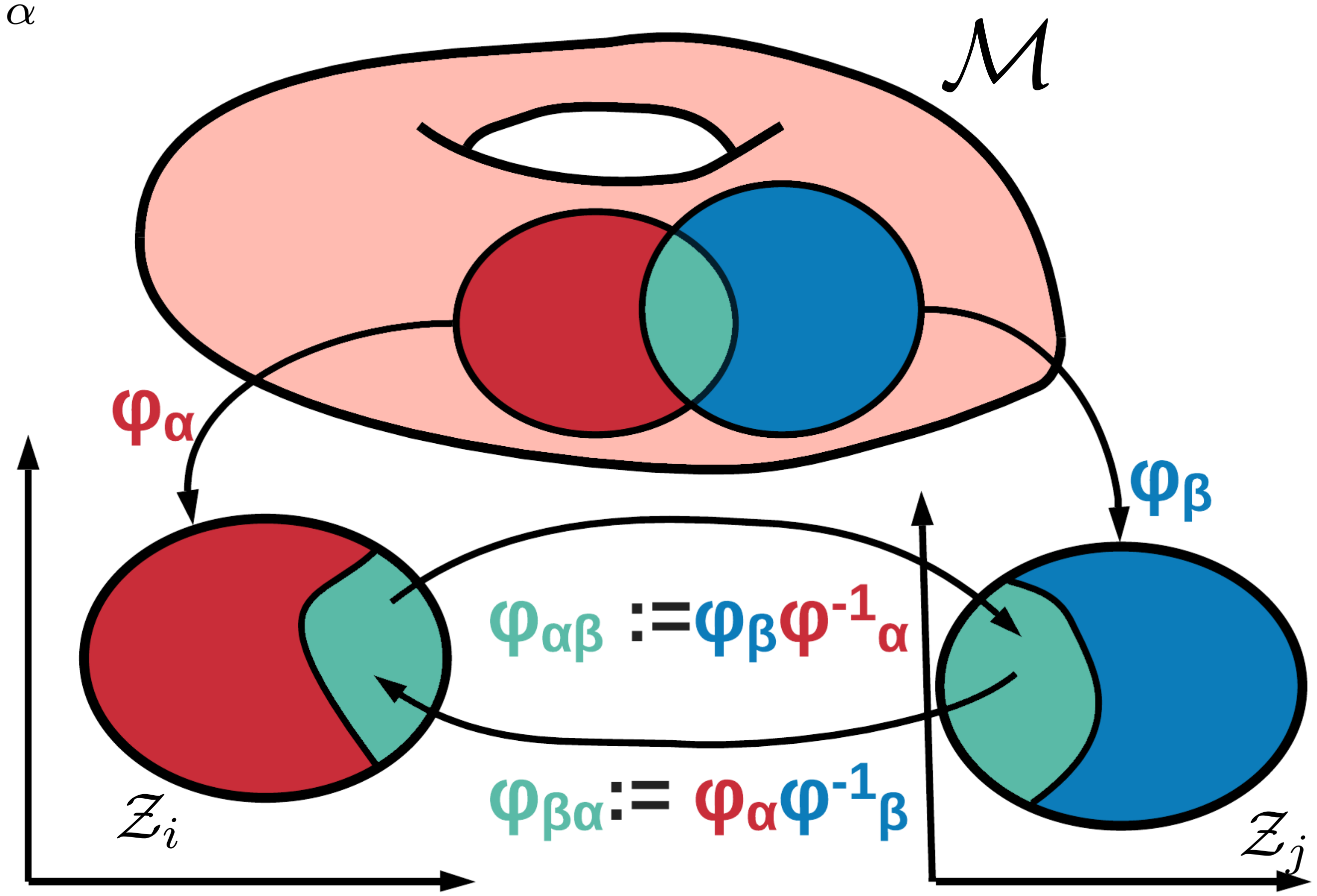
A manifold with a small reach can "bend" faster than the one with a large reach. For example, a plane has a reach equal to infinity. A hyper-sphere with radius r has a reach r.

Theorem 1. *(Schonsheck-Chen-Lai) Let* $\mathcal{M}$ *be a d-dimensional compact manifold. If an auto-encoder* $(\mathcal{Z}; \boldsymbol{E}, \boldsymbol{D})$ *of* $\mathcal{M}$ *is an* ϵ*-faithful representation with* $\epsilon < \tau(\mathcal{M})$*, then* $\mathcal{Z}$ *and* $\boldsymbol{D}(\mathcal{Z})$ *must be homeomorphic to* $\mathcal{M}$*. Particularly, a d-dimensional compact manifold with non-contractible topology can not be* ϵ*-faithfully represented by a plain auto-encoder with a latent space* $\mathcal{Z}$ *being a d-dimensional simply connected domain in* $\mathbb{R}^d$.

Differential manifold point of view

A manifold is a topological space locally homeomorphic to a Euclidean domain.

- Charts $\{(\mathcal{M}_\alpha, \phi_\alpha)\}_\alpha$ satisfying $\mathcal{M} = \bigcup_\alpha \mathcal{M}_\alpha$
- Coordinate map: $\phi_\alpha : \mathcal{M}_\alpha \to \mathcal{Z}_\alpha$
- Transition functions:
$$\phi_{\alpha\beta} : \phi_\alpha(\mathcal{M}_\alpha \cap \mathcal{M}_\beta) \to \phi_\beta(\mathcal{M}_\alpha \cap \mathcal{M}_\beta)$$



Machine learning:

- $\mathcal{M}$: data manifold
- $\mathcal{Z}_\alpha$: Latent space
- ϕ_α: Encoders E_α approximated by DNNs
- ϕ_α^{-1}: Decoder D_α approximated by DNNs

[Partition of Unity]

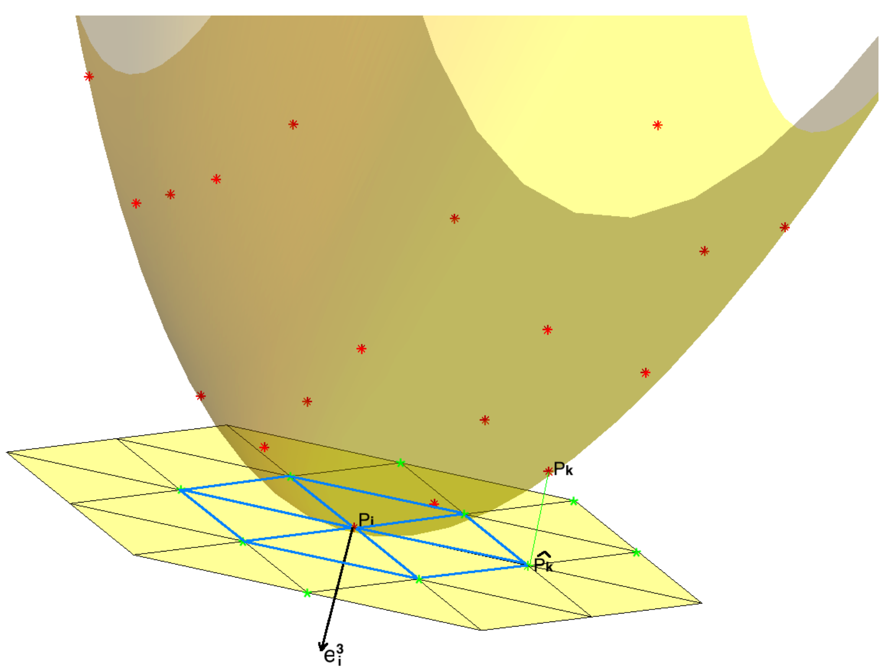
1. Only a finite number of the functions in $\{\rho_k\}_{k\in\mathcal{K}}$ are nonzero near $\mathbf{x}$ and $\sum_{k\in\mathcal{K}} \rho_k(\mathbf{x}) = 1$.
2. Assemble from local chart: $f(\mathbf{x}) = \sum_{k\in\mathcal{K}} \rho_k(\mathbf{x}) f(\mathbf{x})$

Universal Manifold Approximation

Theorem 1 (Schosheck-Chen-Lai). *Consider a d-dimensional compact data manifold $\mathcal{M} \subset \mathbb{R}^m$ with reach τ. Let $X = \{x\}_{i=1}^n$ be a training data set drawn uniformly randomly on $\mathcal{M}$. For any $0 < \epsilon < \tau/2$, if $|X| \approx O(-d\epsilon^{-d} \log \epsilon)$ then there exists a Chart Auto-encoder $(\boldsymbol{E}, \boldsymbol{D}) = \arg\min_{\boldsymbol{E},\boldsymbol{D}} f(\Theta; X) = \frac{1}{n} \sum_i \|x_i - \boldsymbol{D} \circ \boldsymbol{E}(x_i)\|^2$ ϵ-faithfully representing $\mathcal{M}$, namely*

$$\sup_{x \in \mathcal{M}} \|x - \boldsymbol{D} \circ \boldsymbol{E}(x)\| \leq \epsilon.$$

Moreover, the encoder $\boldsymbol{E}$ and the decoder $\boldsymbol{D}$ has at most $O(Lmd\epsilon^{-d-d^2/2}(-\log^{1+d/2} \epsilon))$ parameters and $O(-d^2 \log_2 \epsilon/2)$ layers.



Step 1. $X = \{x_i\}_{i=1}^n$ *forms* $\epsilon/2$*-dense* ($\epsilon < \tau/2$) *sampling if* $|X| >= O\left(-d\epsilon^{-d} \log \epsilon\right)$. *[Niyogi-Smale-Weinberger'08]*

Step 2. *Representing simplicial maps locally. Consider a geodesic neighborhood* $\mathcal{M}_r(p) = \{x \in \mathcal{M} \mid d(p, x) < r\}$ *around* $p \in \mathcal{M}$. *For any* $0 < \epsilon < \tau(\mathcal{M})$, *if* $X = \{x_i\}_{i=1}^n$ *is an* $\epsilon/2$*-dense sample drawn uniformly randomly on* $\mathcal{M}_r(p)$, *then there exists an auto-encoder* $(\mathcal{Z}, \boldsymbol{E}, \boldsymbol{D}) = \arg\min_{\boldsymbol{E},\boldsymbol{D}} \sum \|x_i - \boldsymbol{D} \circ \boldsymbol{E}(x_i)\|^2$ *satisfying* $\sup_{x \in \mathcal{M}_r(p)} \|x - \boldsymbol{D} \circ \boldsymbol{E}(x)\| \leq \epsilon$.

Step 3. *Gluing local results through partition of unity.*

Consider a training data set $\mathcal{S} = \{(\boldsymbol{x}_i, \mathbf{v}_i)\}_{i=1}^n$ where the $\mathbf{v}_i$'s are i.i.d. samples from a probability measure on $\mathcal{M}$, and

$$\mathbf{x}_i = \mathbf{v}_i + \mathbf{w}_i$$

are perturbed from the $\mathbf{v}_i$'s with independent random normal noise $\mathbf{w}_i \in T^{\perp}_{\mathbf{v}_i}\mathcal{M}$ (the normal space of $\mathcal{M}$ at $\mathbf{v}_i$) satisfying $\|\mathbf{w}_i\|_2 \leq q < \tau$. We denote the distribution of all $\mathbf{x}_i$ by γ.

Our goal is to learn an encoder $\widehat{\mathbf{E}} : \mathcal{M}(q) \to \mathbb{R}^{O(d)}$ and the corresponding decoder $\widehat{\mathbf{D}} : \mathbb{R}^{O(d)} \to \mathbb{R}^D$ by minimizing the empirical mean squared loss

$$(\widehat{\mathbf{D}}, \widehat{\mathbf{E}}) = \underset{\mathbf{D}\in\mathcal{F}_{\mathrm{NN}}^{\mathbf{D}}, \mathbf{E}\in\mathcal{F}_{\mathrm{NN}}^{\mathbf{E}}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{D}\circ\mathbf{E}(\mathbf{x}_i)\|_2^2,$$

for some network function classes $\mathcal{F}_{\mathrm{NN}}^{\mathbf{E}}$ and $\mathcal{F}_{\mathrm{NN}}^{\mathbf{D}}$ given by properly designed network architectures.

Extension: Generalization bound under noisy input (with H. Liu, A. Havrilla, W. Liao)

Theorem (Informal). Suppose the encoder $\mathscr{E}: \mathbb{R}^D \to \mathbb{R}^{O(d)}$ and decoder $\mathscr{D}: \mathbb{R}^{O(d)} \to \mathbb{R}^D$ network architectures are properly set. Let $\widehat{\mathscr{E}}$ and $\widehat{\mathscr{D}}$ be the global minimizer of the empirical risk. We have

$$\mathbb{E}_{\mathcal{S}}\mathbb{E}_{\mathbf{x}\sim\gamma}\|\widehat{\mathscr{D}} \circ \widehat{\mathscr{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2 \leq CD^2 \log^2 D n^{-\frac{2}{d+2}} \log^4 n$$

where C is a constant independent of n and D, and number of layers $O(\log^2 n + \log D)$, width $O(Dn^{\frac{d}{d+2}})$ and number parameters $O(Dn^{\frac{d}{d+2}} \log^2 n + D \log D)$

- Given accuracy ϵ, data size $n \sim \epsilon^{-(d+2)/2}$, network parameters $O(\epsilon^{-d/2})$
- Robustness noise on normal directions
- For noise with tangential components with 2nd moment bounded by σ^2. We can have

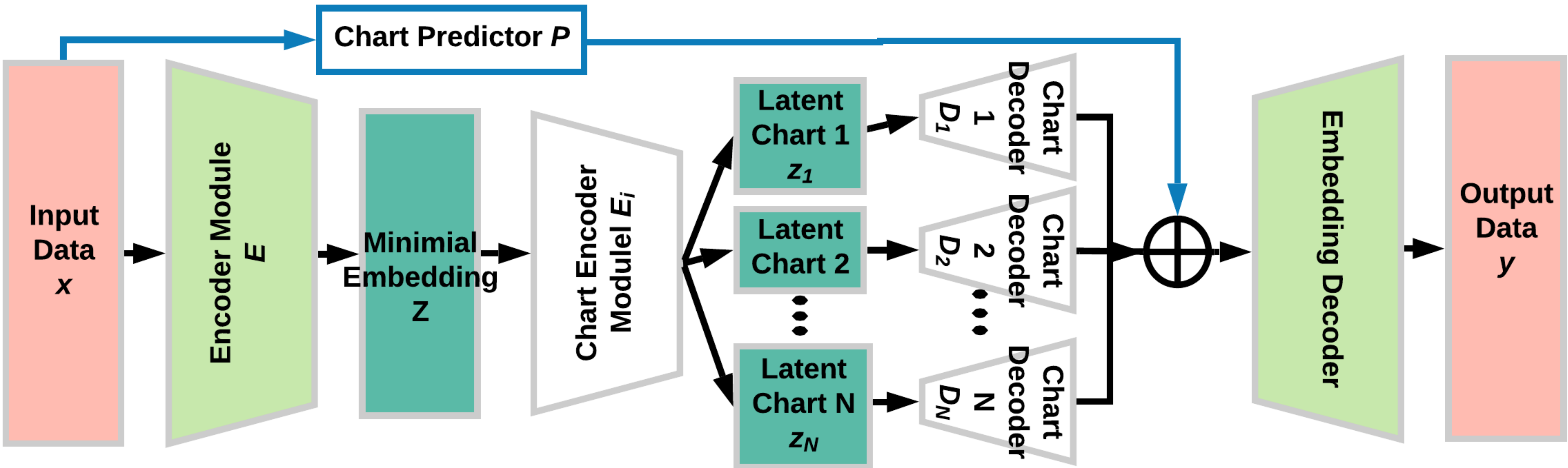
$$\mathbb{E}_{\mathcal{S}}\mathbb{E}_{\mathbf{x}\sim\gamma}\|\widehat{\mathbf{D}} \circ \widehat{\mathbf{E}}(\mathbf{x}) - \mathbf{v}\|_2^2 \leq C(D^2 \log^3 D) n^{-\frac{2}{d+2}} \log^4 n + C_1\sigma^2$$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\mathbb{E}_{\mathbf{x}\sim\gamma}\left[\|\widehat{\mathbf{D}} \circ \widehat{\mathbf{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2\right] \\
&= \underbrace{2\mathbb{E}_{\mathcal{S}}\left[\frac{1}{n}\sum_{i=1}^{n}\|\widehat{\mathbf{D}} \circ \widehat{\mathbf{E}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2\right]}_{\mathrm{T}_1} + \underbrace{\mathbb{E}_{\mathcal{S}}\mathbb{E}_{\mathbf{x}\sim\gamma}\left[\|\widehat{\mathbf{D}} \circ \widehat{\mathbf{E}}(\mathbf{x}) - \pi(\mathbf{x})\|_2^2\right] - 2\mathbb{E}_{\mathcal{S}}\left[\frac{1}{n}\sum_{i=1}^{n}\|\widehat{\mathbf{D}} \circ \widehat{\mathbf{E}}(\mathbf{x}_i) - \pi(\mathbf{x}_i)\|_2^2\right]}_{\mathrm{T}_2}.
\end{aligned}$$

Bound Approximation error

Bound Variance through the covering number

Network Architecture: A unsupervised method



- Chart prediction $\{p_\alpha\}$ is approximated by a DNN
- Write $y_\alpha = \mathbf{D} \circ \mathbf{D}_\alpha \circ \mathbf{E}_\alpha \circ \mathbf{E}(x)$, define $e_\alpha = \|x - y_\alpha\|^2$ and an internal label $\ell_\alpha = \text{softmax}(-e_\alpha)$. Then the *Chart-Prediction Loss* is given by:

$$\mathcal{L}_{CP}(x, \Theta) := \left(\min_\alpha e_\alpha\right) - \sum_{\beta=1}^{N} \ell_\beta \log(p_\beta)$$

Regularization and pre-training

Lipschitz regularization Denoting the weights of the k^{th} layer of E_α as W_α^k, we propose the following regularization on the decoder functions for a K-layer network:

$$\mathcal{R}_{Lip} := \max_\alpha \prod_{k=1}^{K} ||W_\alpha^k||_2 + \frac{1}{N} \sum_{\beta=1}^{N} \prod_{k=1}^{K} ||W_\beta^k||_2$$

Pre-training

- Applying furthest point sampling (FPS) scheme to select N data points. Then we assign each of these data points to a decoder and train each one to reconstruct.
- Train the encoder such that x_α is at the center of the chart space U_α.
- We further define the chart prediction probability as the categorical distribution and use it to pre-train the chart predictor.

$$\mathcal{L}_{init}(x_\beta) := \|x_\beta - \mathbf{D}_\beta \circ \mathbf{E}_\beta \circ \mathbf{E}(x_\beta)\|^2 + \|\mathbf{E}_\beta \circ \mathbf{E}(x_\beta) - [.5]^d\|^2 + \sum_{\alpha=1}^{N} \delta_{\alpha\beta} \log(p_\alpha).$$

Illustrative example: Effects of Lipschitz Regularization

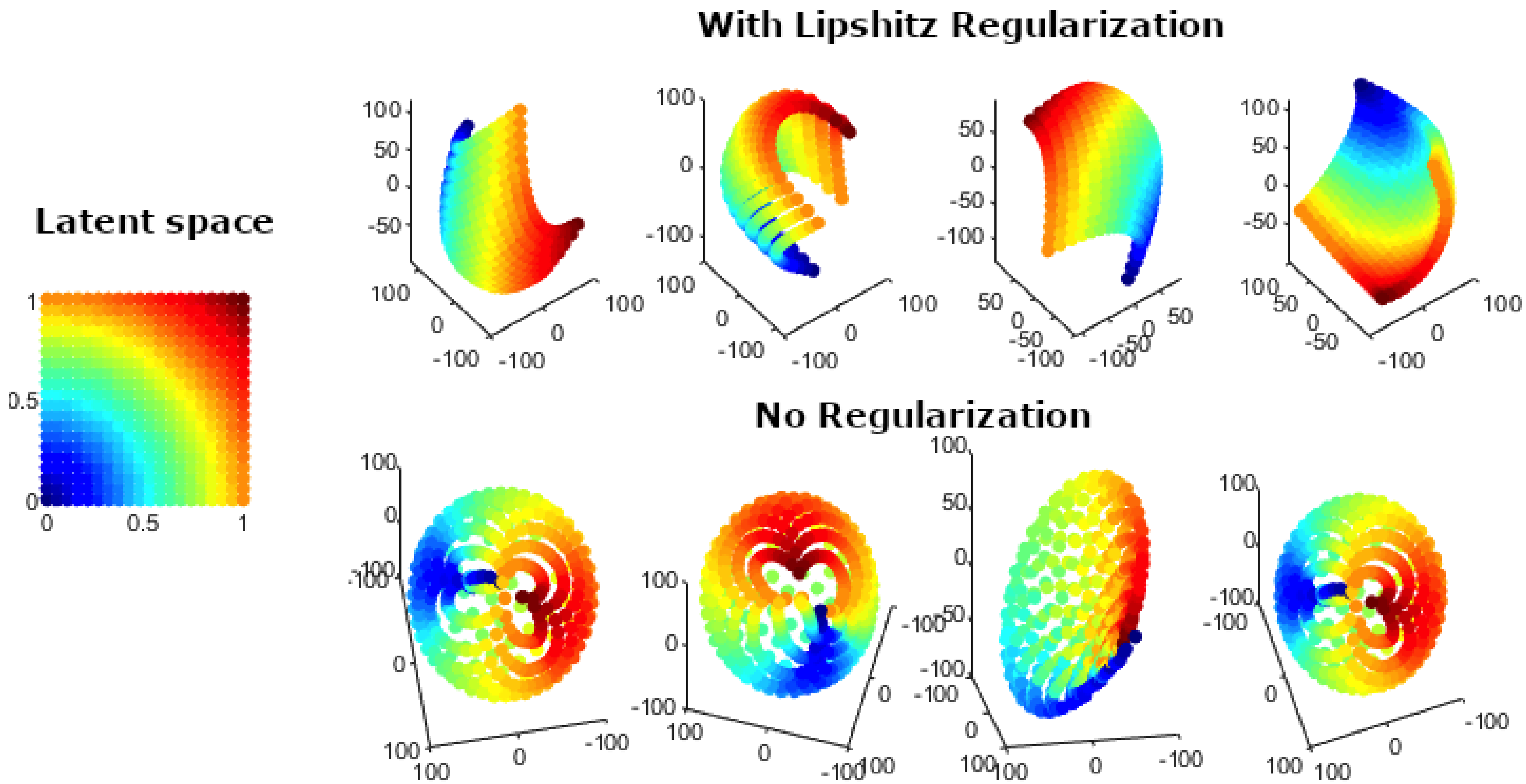


Figure 1: Left: Chart latent space. Top: Model with Lipschitz regularization. Bottom: Model without Lipschitz regularization.

Illustrative example: Learning charts

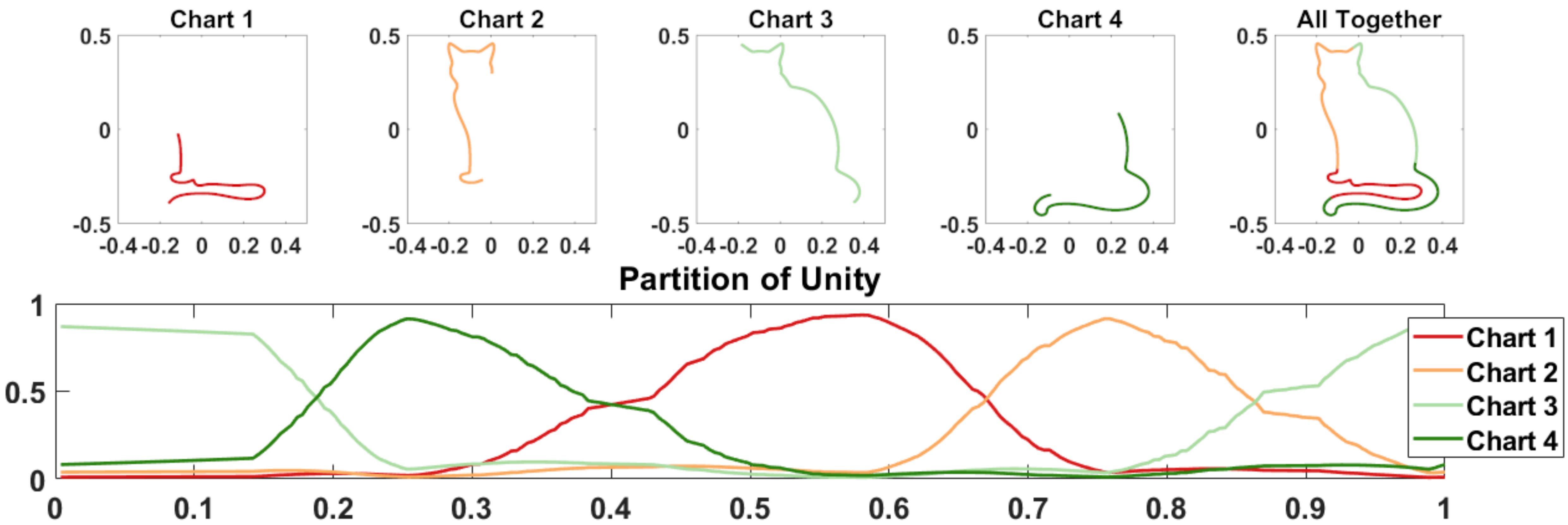


Figure 1: Top: The first four are individual charts and the last one is a concatenation of them by taking the max of chart probabilities p_α. Bottom: Variation of p_α for each training point on the manifold.

Illustrative example: Automatic Chart Removal

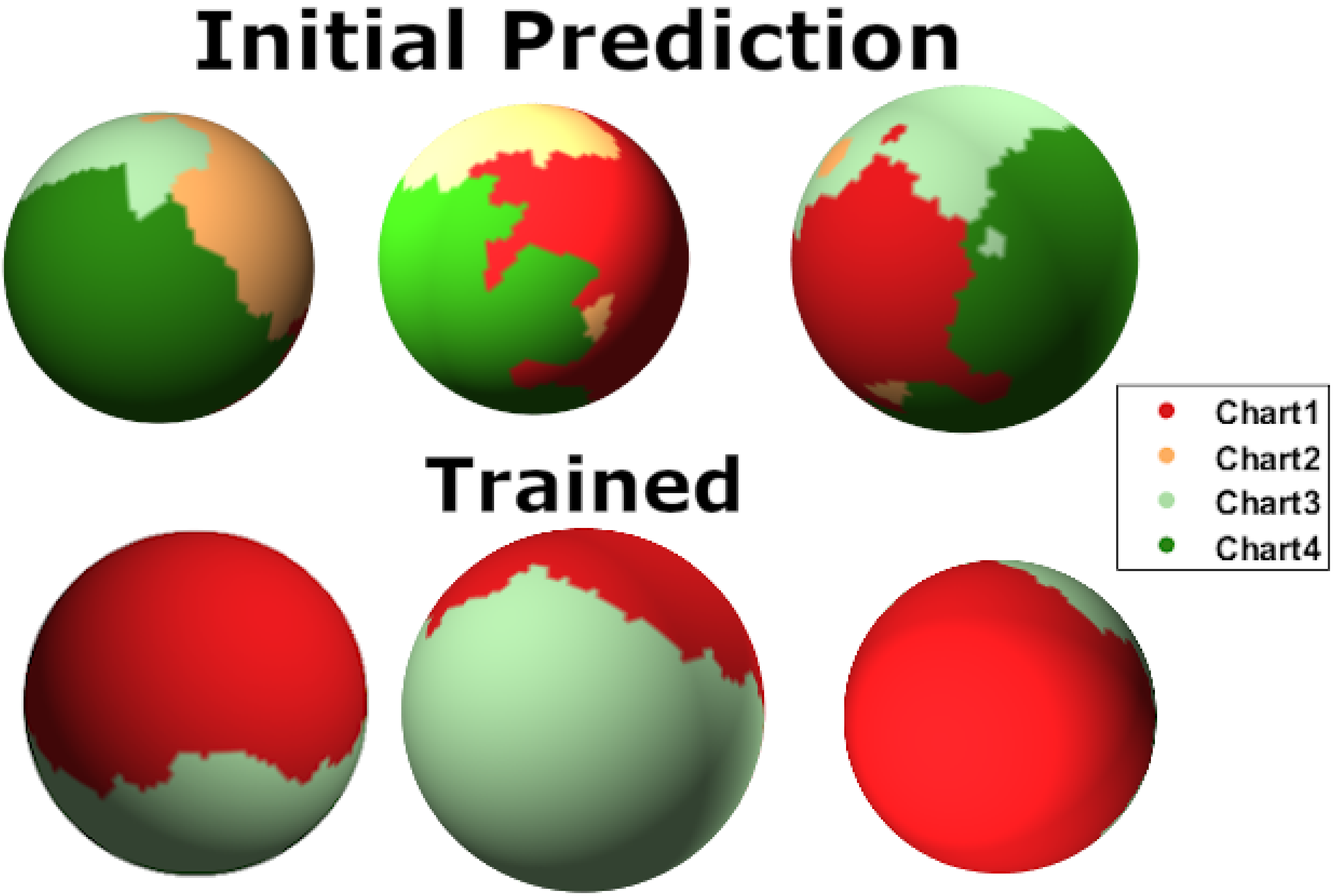


Figure 1: Top: Pre-trained charts. Bottom: Final charts after training.

Illustrative example: VAEs do not generalize for double torus

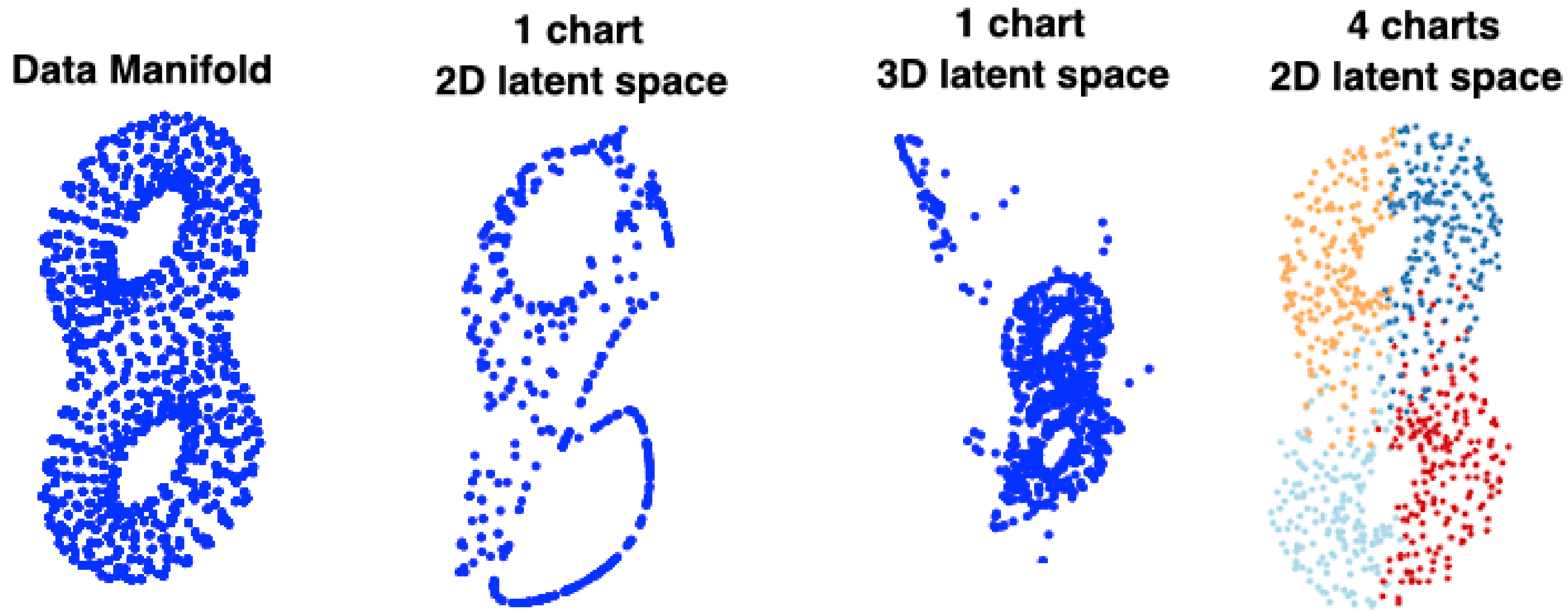


Figure 1: Left: Data on a double torus. Middle two: Data auto-encoded to a flat latent space. Right: Data auto-encoded to a 4-chart latent space.

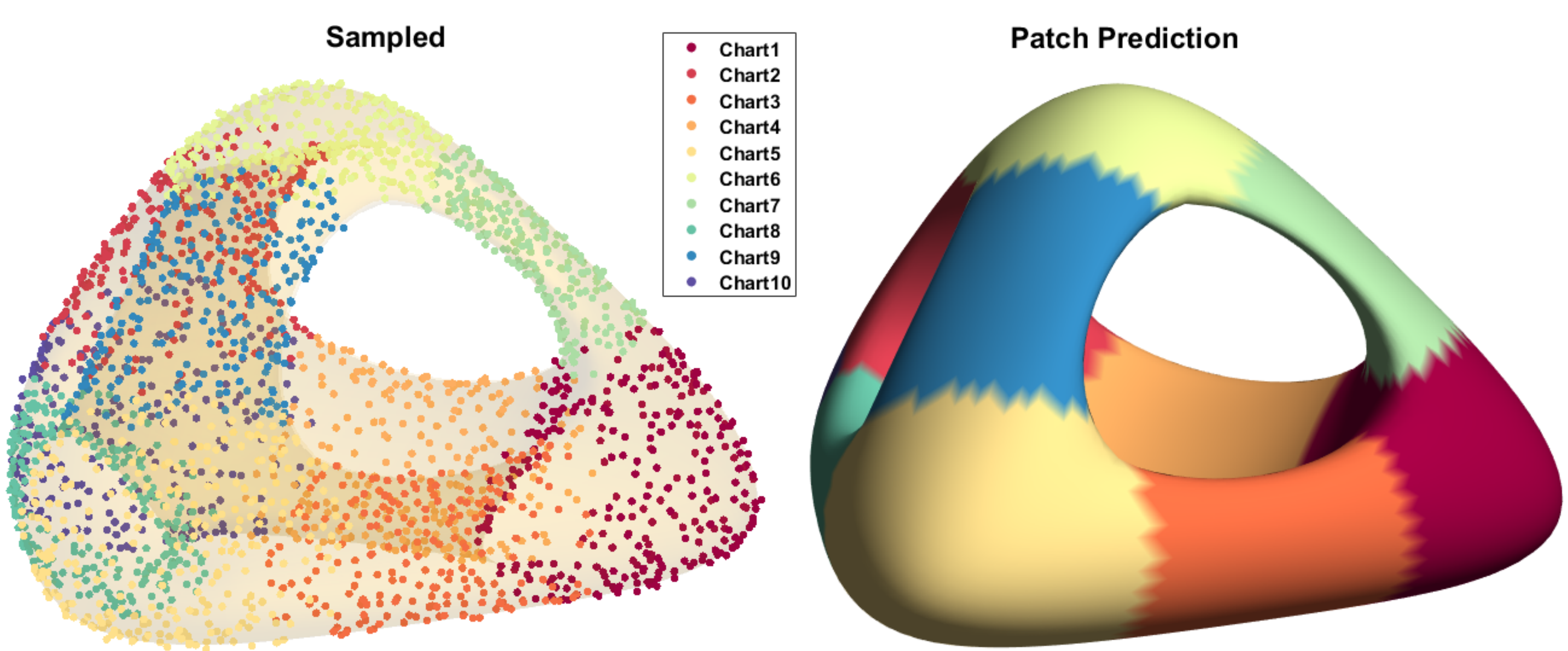
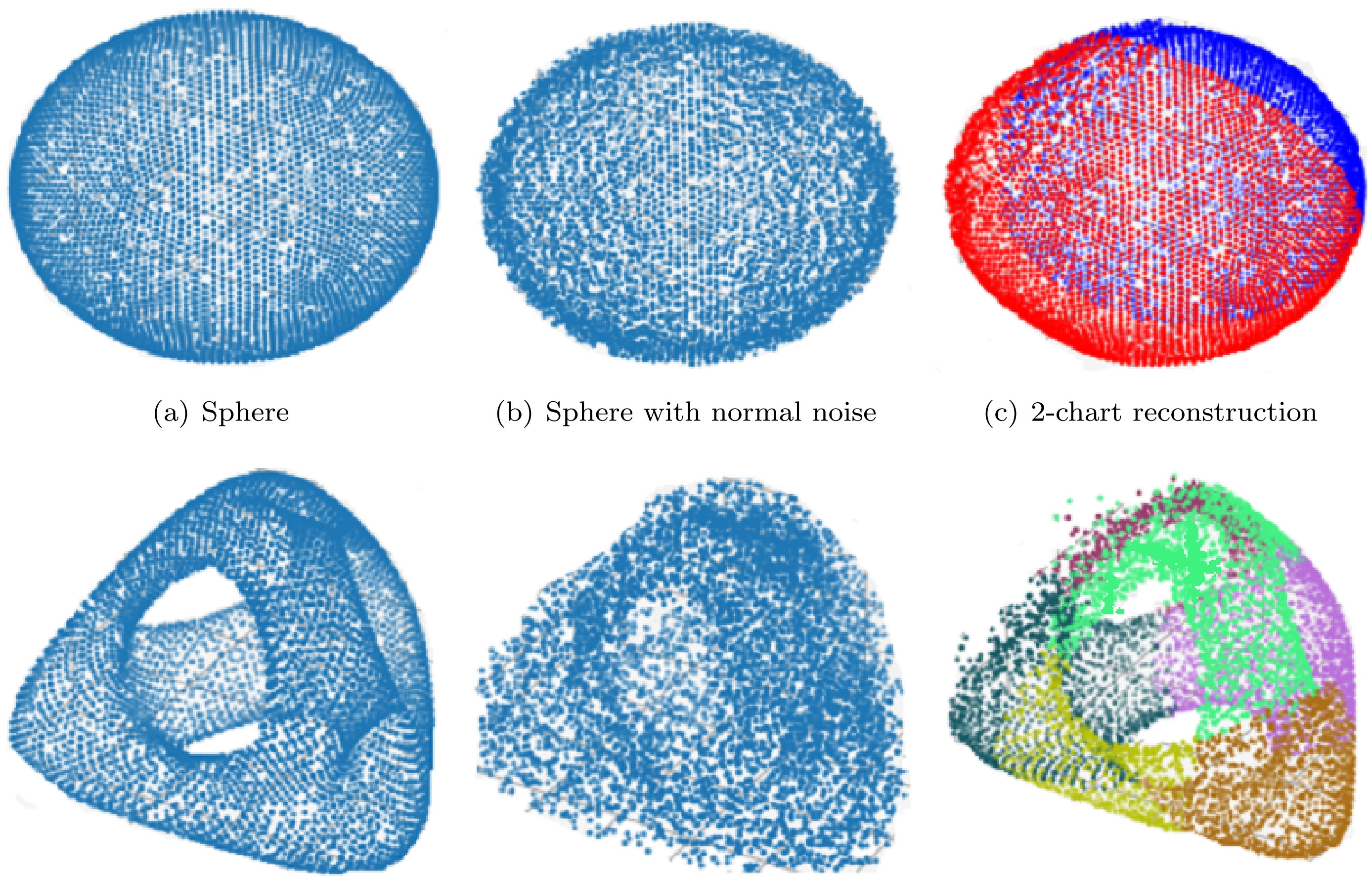


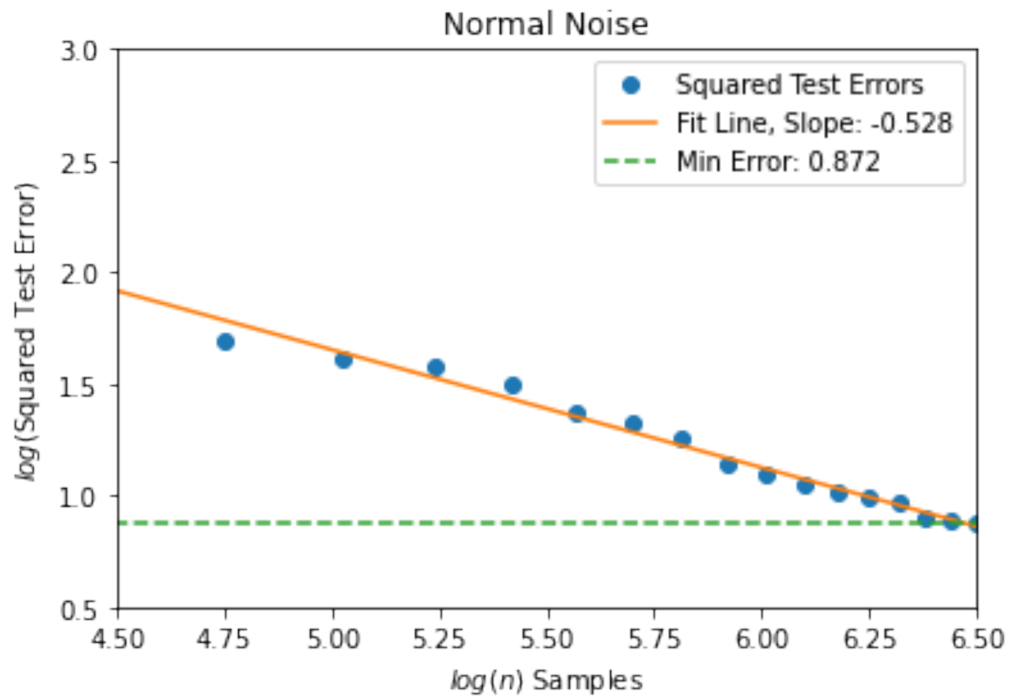
Figure 1: Left: Points sampled from high probability regions. Right: Charts after taking max.

Robust to noise on normal directions



(a) Sphere (b) Sphere with normal noise (c) 2-chart reconstruction

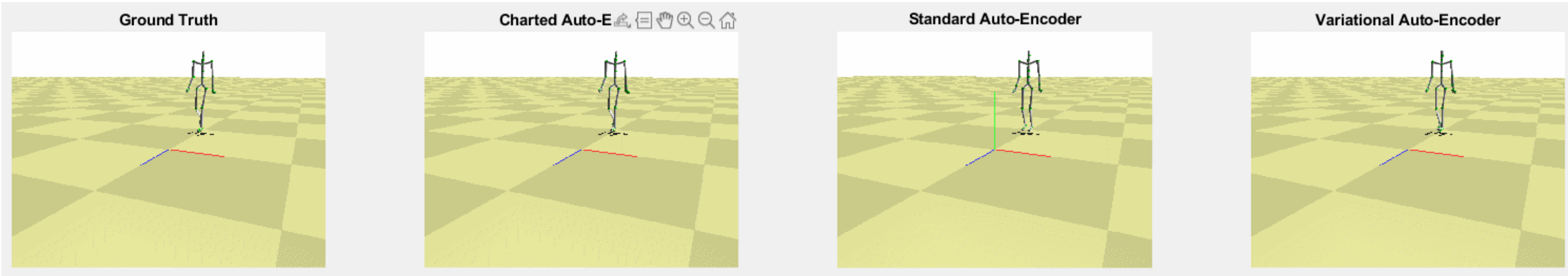
(d) Genus-3 pyramid (e) Genus-3 with normal noise (f) 8-chart reconstruction

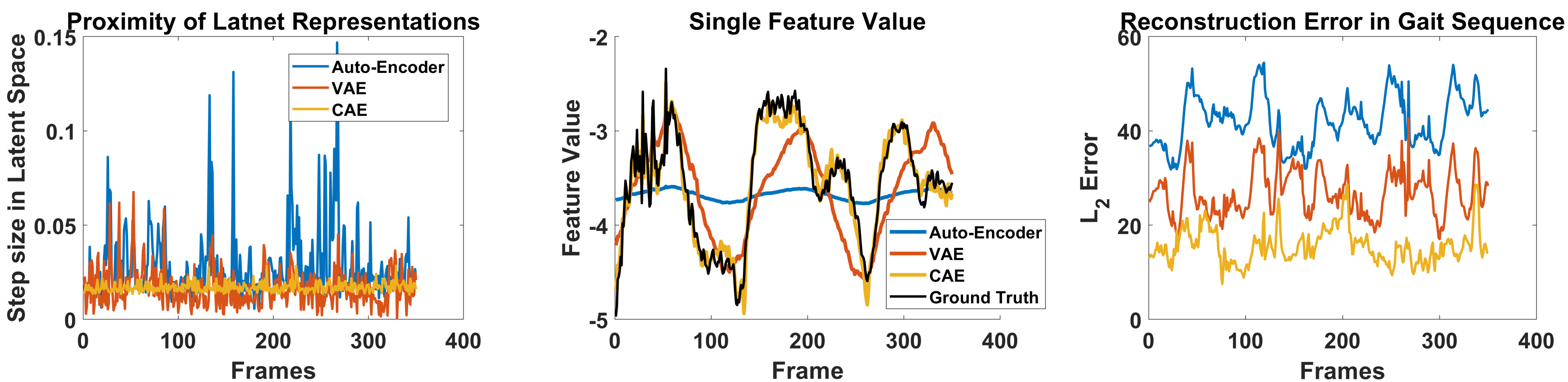


Pyramid

$$\mathbb{E}_{\mathcal{S}}\mathbb{E}_{\mathbf{x}\sim\gamma}\|\widehat{\mathscr{D}}\circ\widehat{\mathscr{E}}(\mathbf{x})-\pi(\mathbf{x})\|_2^2 \le CD^2\log^2 D n^{-\frac{2}{d+2}}\log^4 n$$

Human Motion Data





(a) (b) (c)

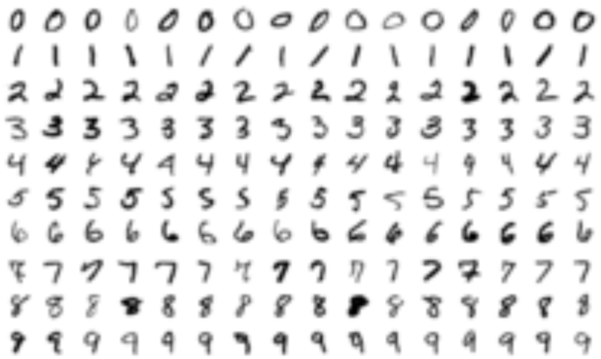
Figure 1: Auto-encoding human motion sequence. (a): Distance between consecutive frames in the latent space. (b): Value of a single feature. (c): Reconstruction error for all features.

Comparisons

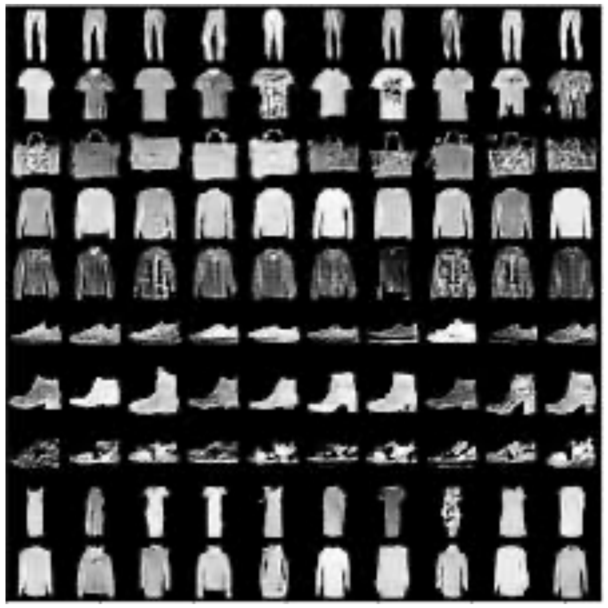
Reconstruction Error $\mathcal{E}_{recon} := \frac{1}{n}\sum_{x\in D_{Test}} \|x-y\|^2$.

Unfaithfulness Let $\{z_i\}_{i=1}^{\ell} \in \mathcal{Z}$. The unfaithfulness is $\mathcal{E}_{unfaithful} = \frac{1}{\ell}\sum_{i=1}^{\ell} \min_{x\in D_{train}} \|x - \mathbf{D}(z_i)\|^2$.

Coverage Let $\ell^* = |\{x^* \mid x^* = \arg\min_{x\in D_{train}} \|x - \mathbf{D}(z_i)\|^2\}|$. Then, we define the coverage $\mathcal{E}_{coverage} = \ell^*/\ell$.



MNIST



FMNIST

Table 1: Reconstruction error and other metrics on MNIST and fashion MNIST.

Model	Charts	Latent Dim	Param.	Recon. Error	Unfaithfulness	Coverage
MNIST						
VAE	1	4	893,028	0.0614 ± .002	0.083 ± .021	0.83 ± .01
	1	64	938,088	0.0512 ± .002	0.070 ± .011	0.94 ± .01
	1	8	2,535,028	0.0564 ± .001	0.085 ± .008	0.91 ± .00
	1	64	2,625,088	0.0391 ± .002	0.081 ± .011	0.96 ± .01
CAE	4	4	601,452	0.0516 ± .001	0.069 ± .019	0.92 ± .01
	4	16	635,196	0.0409 ± .001	0.065 ± .018	0.94 ± .01
	32	16	2,610,120	0.0290 ± .001	0.043 ± .012	0.98 ± .01
	32	32	2,924,808	**0.0289** ± **.002**	**0.045** ± **.011**	**0.98** ± **.01**
FMINST						
VAE	1	8	893,028	0.0575 ± .001	0.016 ± .021	0.80 ± .01
	1	64	938,088	0.0568 ± .003	0.029 ± .034	0.95 ± .01
	1	8	2,535,028	0.0474 ± .001	0.014 ± .008	0.92 ± .00
	1	64	2,625,088	0.0291 ± .006	0.021 ± .011	0.92 ± .01
CAE	4	4	601,452	0.0409 ± .001	0.010 ± .001	0.90 ± .01
	4	16	635,196	0.0301 ± .001	0.010 ± .001	0.90 ± .01
	32	16	2,610,120	0.0190 ± .001	0.016 ± .001	0.97 ± .02
	32	64	3,554,184	**0.0177** ± **.002**	**0.007** ± **.021**	**0.97** ± **.02**

Ongoing/future applications

- **Learning manifolds and functions simultaneously (submitted)**

  Can successfully differentiate nearby but disjoint manifolds and intersecting manifolds with only a small amount of supervision.

- **Operator Learning and Nonlinear Model Reduction (submitted)**

  Theoretical analysis and practical algorithms for operator learning in the latent space.

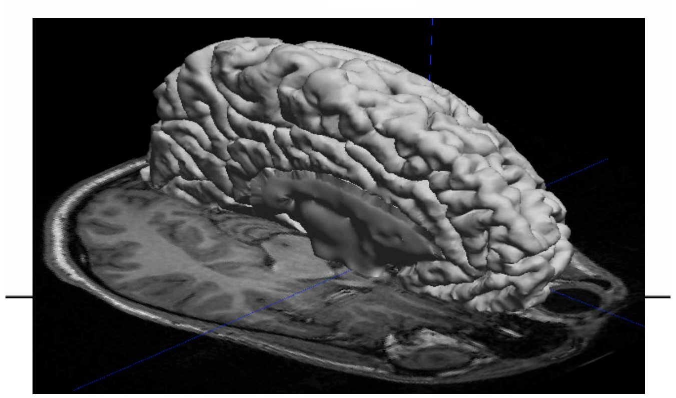
- **Adversarial training (submitted)**

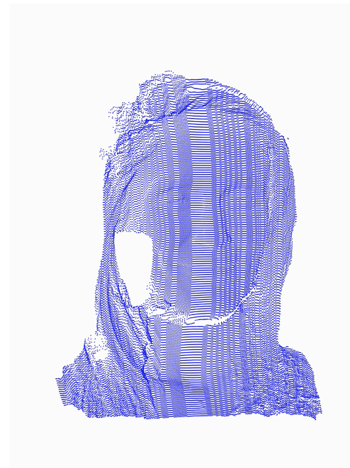
  Enhance the robustness of DNNs by combining with learning data manifold structure

Manifold-structured data in 3D

- 3D modeling
- Image Processing
- Medical Imaging
- ....

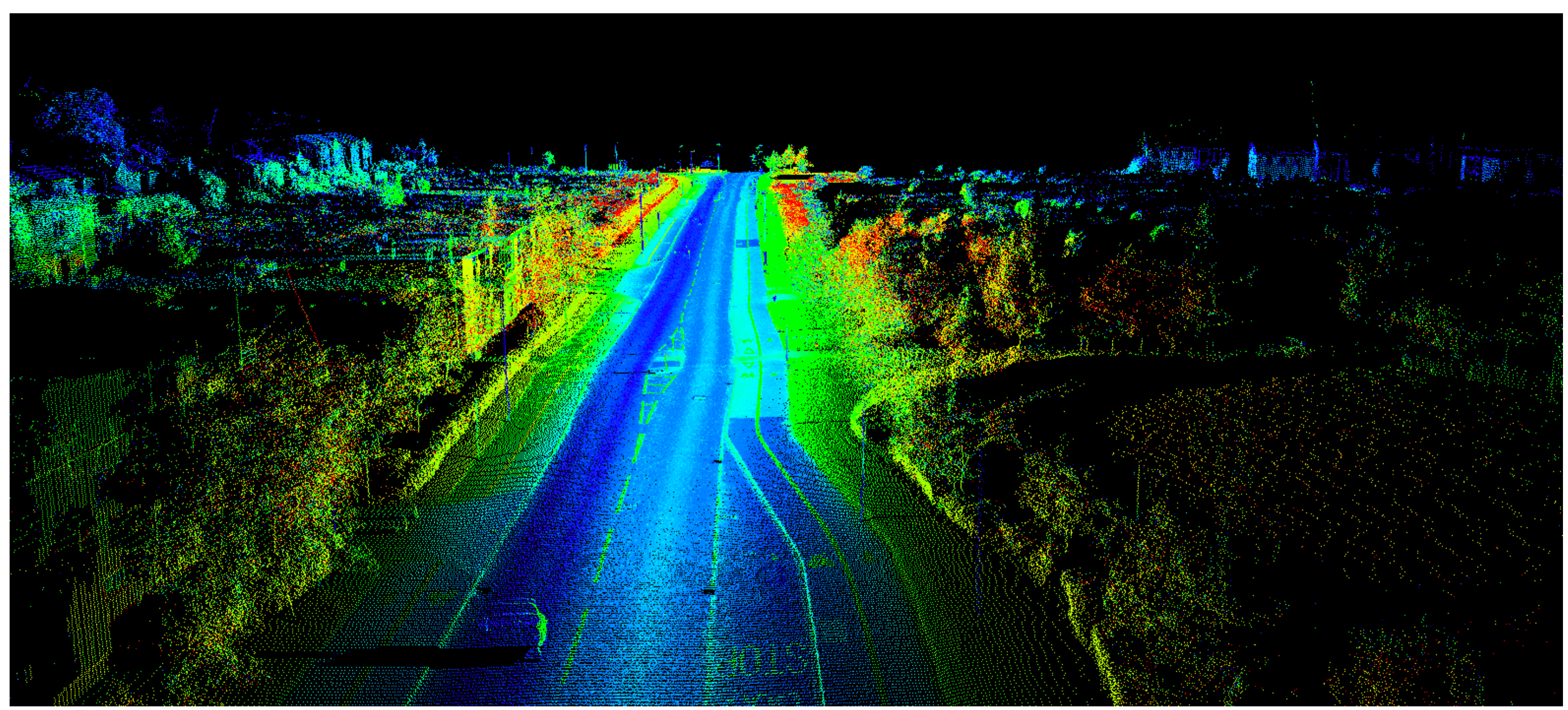
Magnetic Resonance scanner





Data from XYZT Lab

TOSCA Data

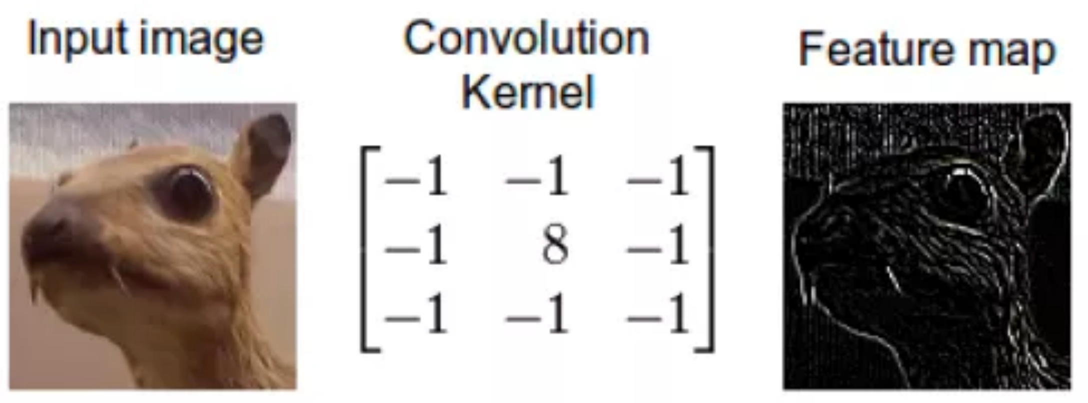


Convolutional Neural Networks

- **Shift invariance is crucial**

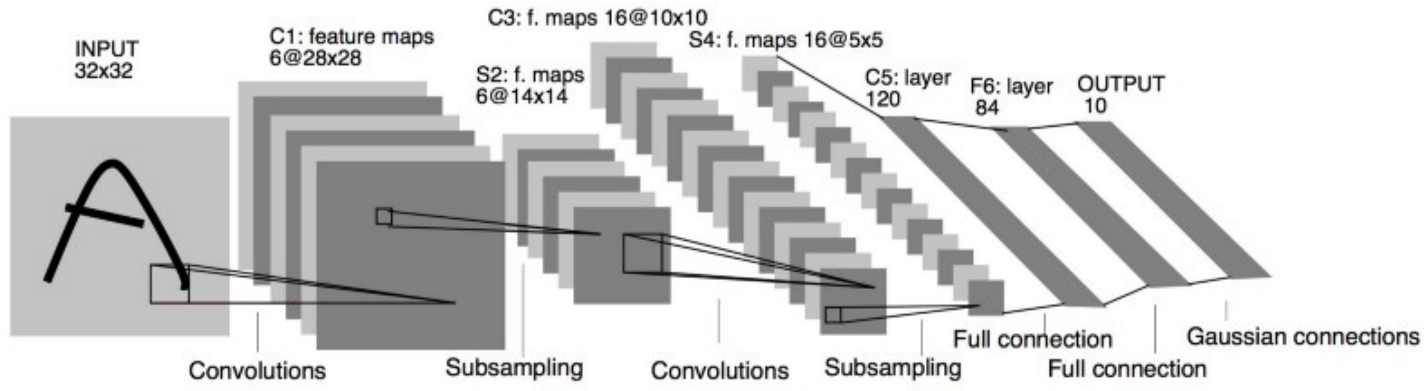
$$(f * k)(x) := \int_{\mathbb{R}^n} k(x-y)f(y)dy$$

$$F_\Theta(x) = f_k(f_{k-1}\cdots f_1(x;w_1);w_2)\cdots;w_k)$$



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

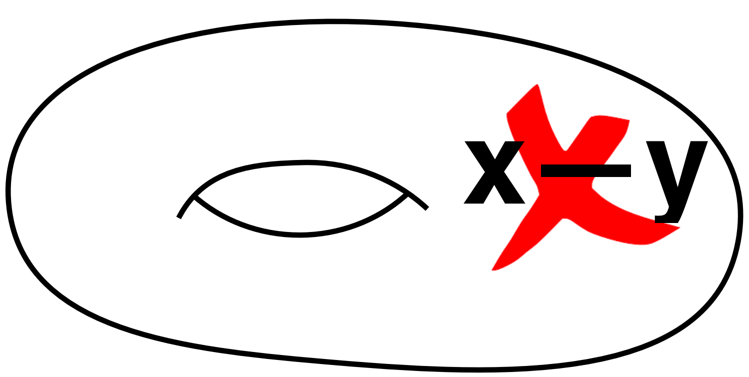
Image from https://timdettmers.com/2015/03/26/convolution-deep-learning/



LeNets [LeCun et al.]

- **Aim at conducting CNN on general manifolds.**

Challenge: a general manifold is not shift invariant



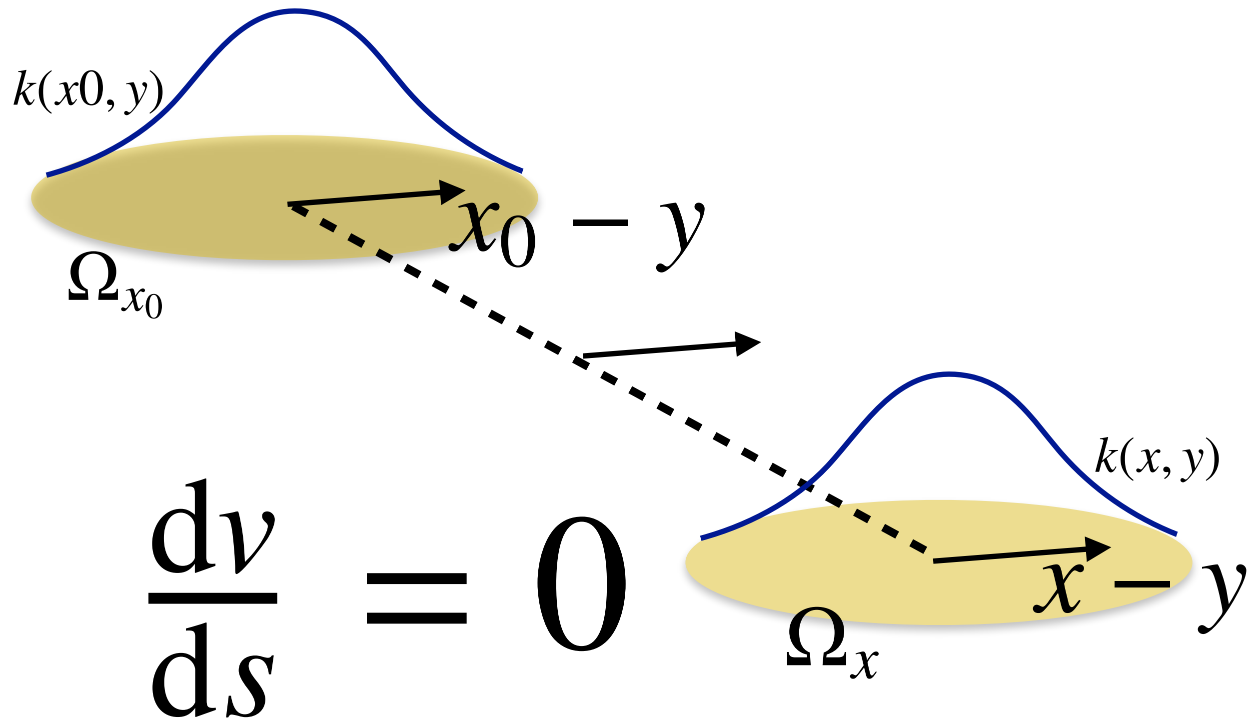
Method	Filter Type	Support	Directional	Transferable	Deformable
Spectral [5]	Spectral	Global	✔	✘	✘
TFG [11]	Spectral	Global	✔	✘	✘
WFT [40]	Spectral	Local	✔	✘	✘
GCNN [31]	Patch	Local	✘	✔	✔
ACNN [3]	Patch	Local	✔	✔	✘
PTC	Geodesic	Local	✔	✔	✔

TABLE 1
Comparison on different generalizations of convolutional operator on general manifolds.

- Group-action based on homogeneous space. G/G_p **[Chakraborty et.al, Tohen et. al. Kondor et. al.]**

Rethink Convolution

$$(f * k)(x) := \int_{\mathbb{R}^n} k(x-y) f(y) dy$$



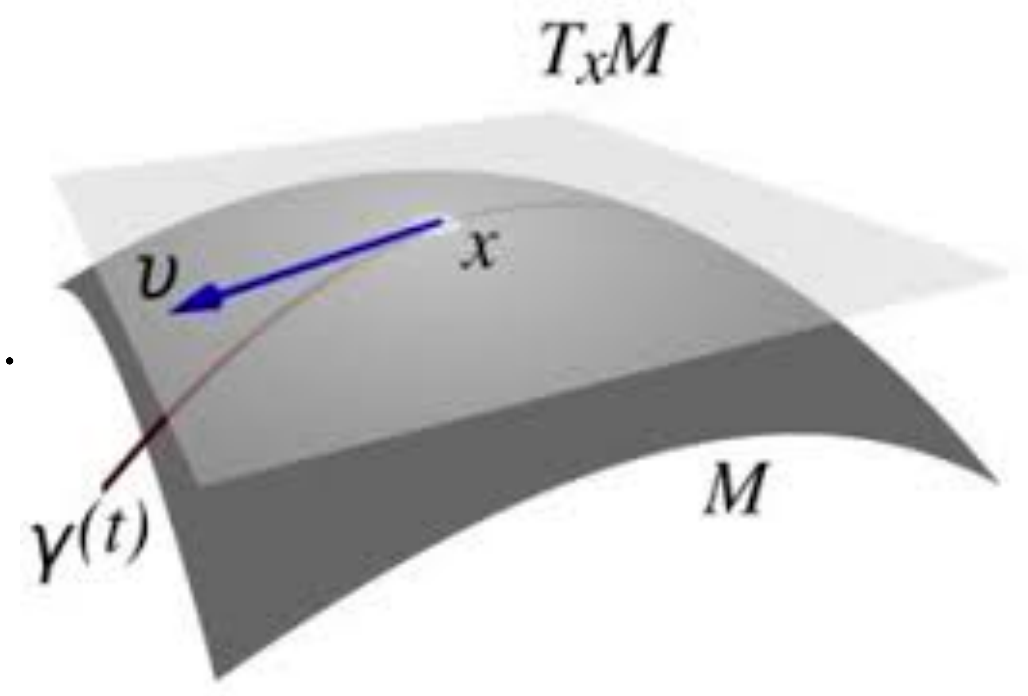
- $(f * k)(x) := \int_{\Omega_x} k(x, y) f(y) dy$
- $(f * k)(x_0) := \int_{\Omega_{x_0}} k(x_0, y) f(y) dy$
- The correspondence of $k(x, y)$ on Ω_x to $k(x_0, y)$ on Ω_{x_0} is provided from the translation map between Ω_{x_0} and Ω_x.

Our method: Convolution on manifold via a parallel transportation [Schonsheck-Dong-Lai]

Exponential Map

A unique geodesic curve γ satisfying $\gamma(0) = x$ and $\gamma'(0) = v$.

$$exp_x(v) = \gamma(1)$$

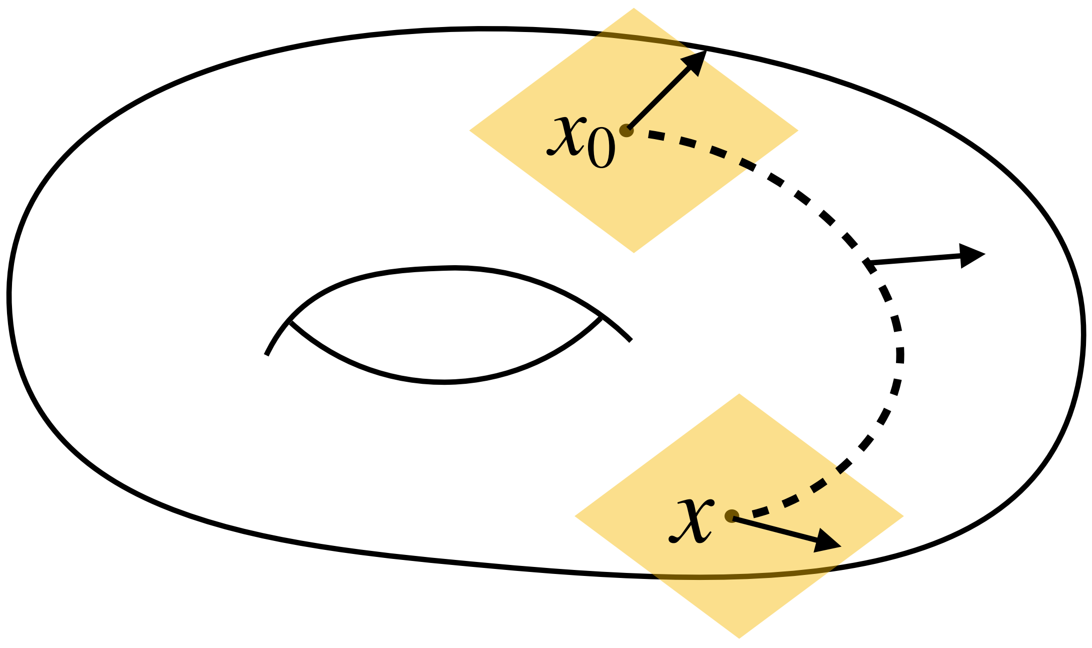


Parallel Transport

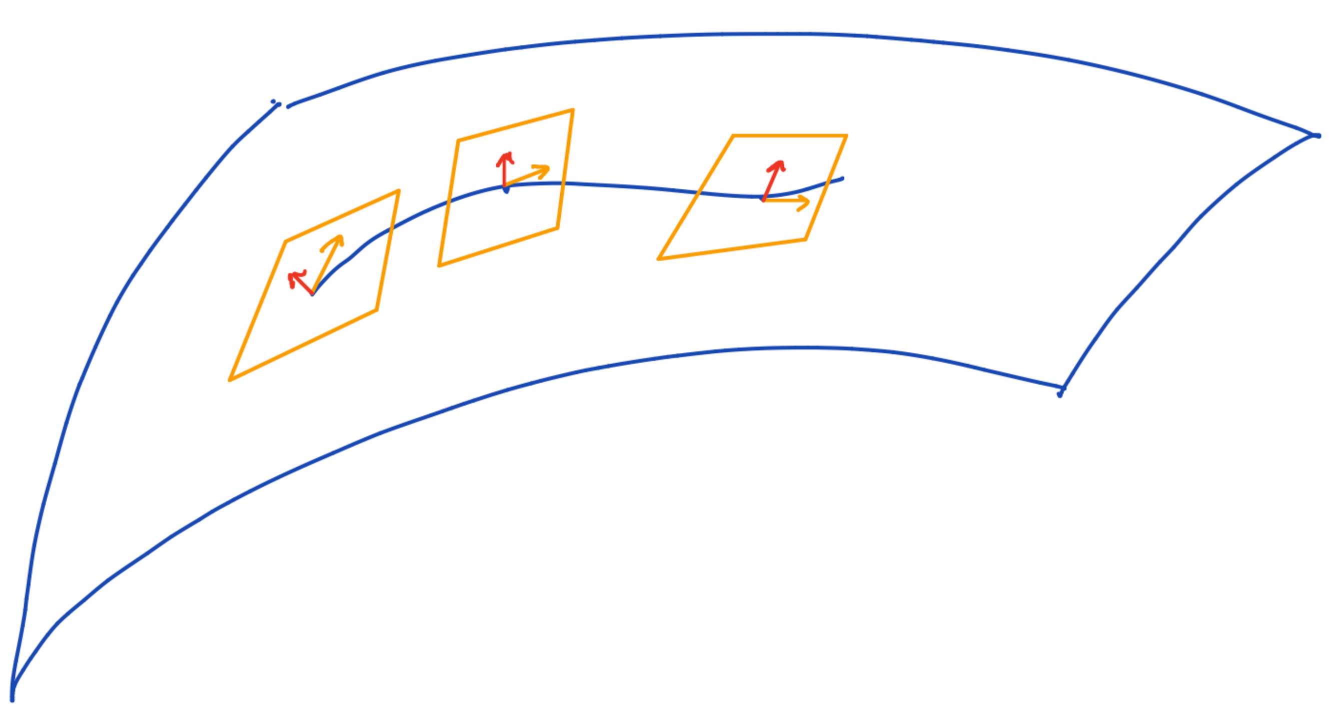
A tangent vector v at $T_{x_0}\mathcal{M}$ can be transported through:

$$\begin{cases} \dfrac{dx^k(t)}{dt} + \dfrac{d\gamma^i}{dt} x^j \Gamma^k_{ij} = 0, \quad k = 1....n, \\ \sum_{i=1}^n x^i(0) e_i = v \end{cases}$$

where Γ^k_{ij} are the Christoffel symbols of the connection.



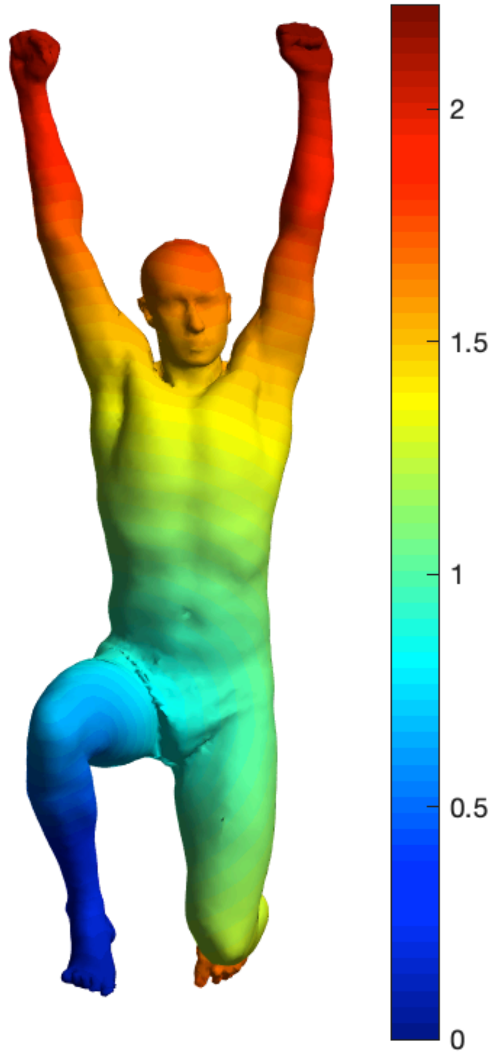
Numerical realization using vector fields from distance functions [Schonsheck-Dong-Lai]

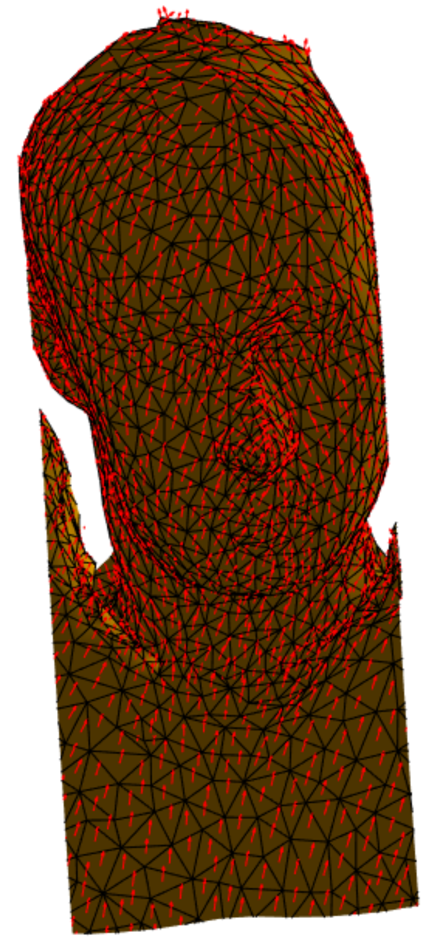


Given smooth vector fields $\{\vec{u}^1, \vec{u}^2\}$, we construct linear transformation among tangent planes $\mathcal{L}(\gamma)_s^t : \mathcal{T}_{\gamma(s)}\mathcal{M} \to \mathcal{T}_{\gamma(t)}\mathcal{M}$ satisfying:

1. $\mathcal{L}(\gamma)$ is smoothly dependent on γ.
2. $\mathcal{L}(\gamma)_s^s = Id$.
3. $\mathcal{L}(\gamma)_u^t \circ \mathcal{L}(\gamma)_s^u = \mathcal{L}(\gamma)_s^t$.

Parallel transport: $\nabla_{\dot{\gamma}} V = \lim_{h\to 0} \frac{1}{h}(\mathcal{L}(\gamma)_0^h(V_{\gamma(0)}) - V_{\gamma(0)})$





The Eikonal equation $|\nabla_{\mathcal{M}} f| = 1$ for local frames

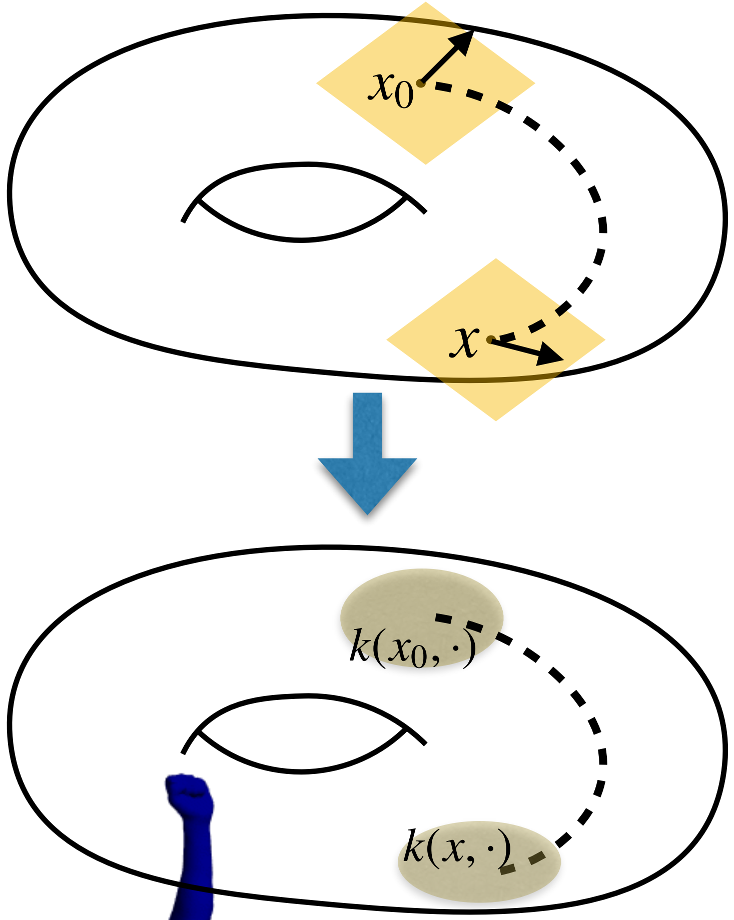
Our method: Convolution on manifold via a parallel transportation

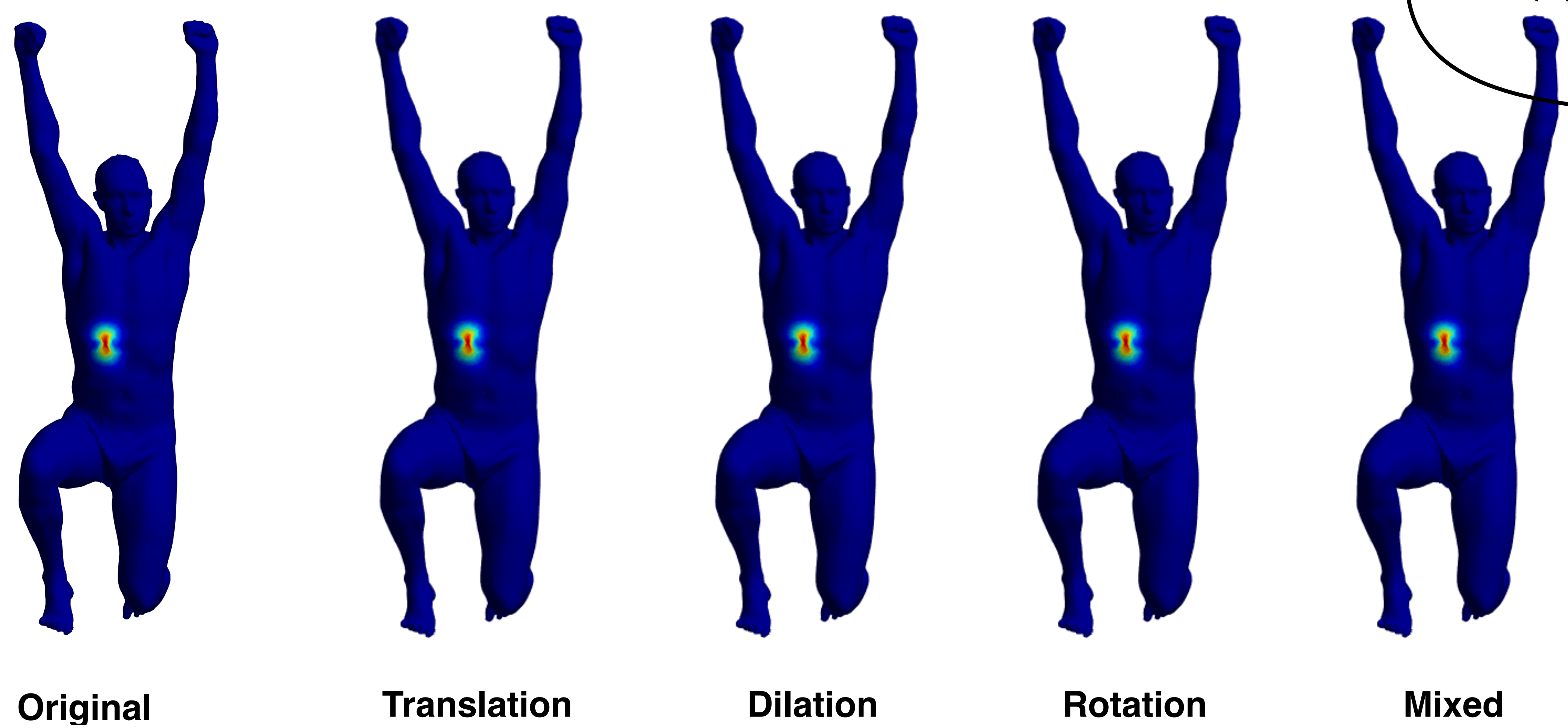
A compactly supported kernel function $k(x_0,\cdot):\mathcal{M}_{x_0,\delta}\to\mathbb{R}$ can be extended on $\mathcal{M}$:

$$k(x,\cdot):\mathcal{M}_{x,\delta}\to\mathbb{R},\quad y\mapsto k\left(x_0, exp_{x_0}\circ \mathrm{PT}^{-1}_{x_0,x}\circ exp_x^{-1}(y)\right)$$

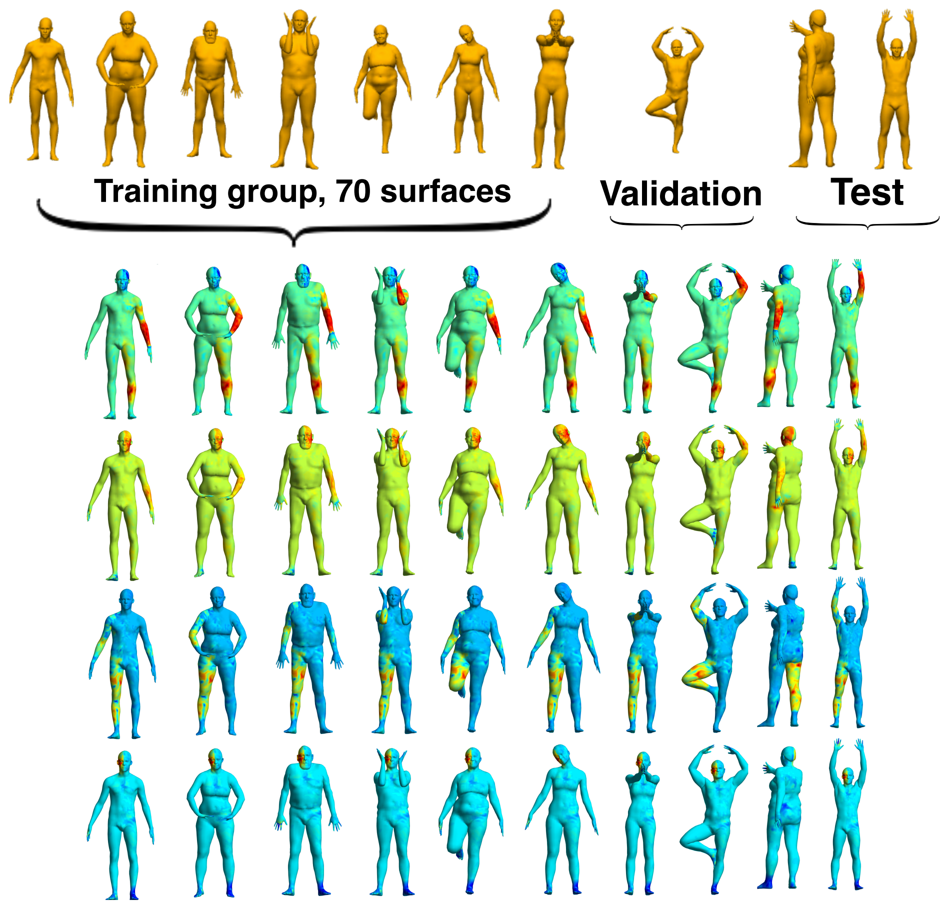
Then, we define convolution as

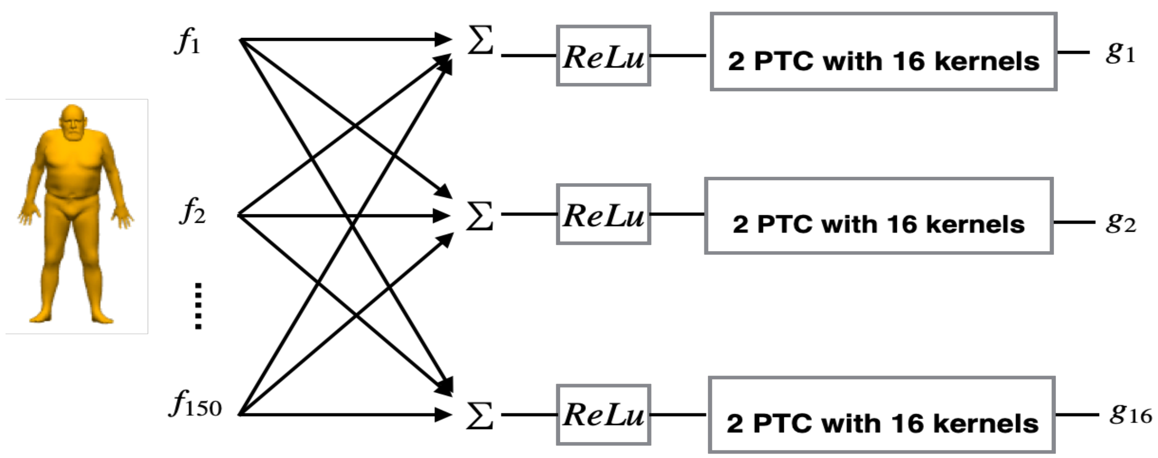
$$f*k(x)=\int_{\mathcal{M}} k(x,y)f(y)\mathrm{d}y=\int_{\mathcal{M}} k\left(x_0, exp_{x_0}\circ \mathrm{PT}^{-1}_{x_0,x}\circ exp_x^{-1}(y)\right)\mathrm{d}y$$

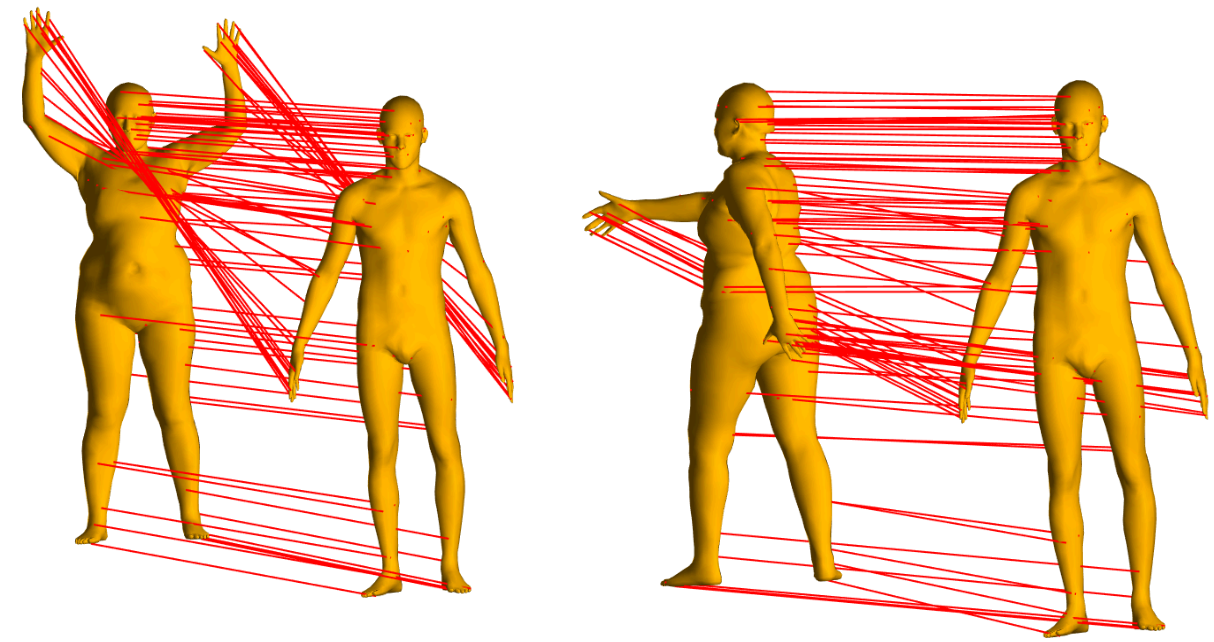




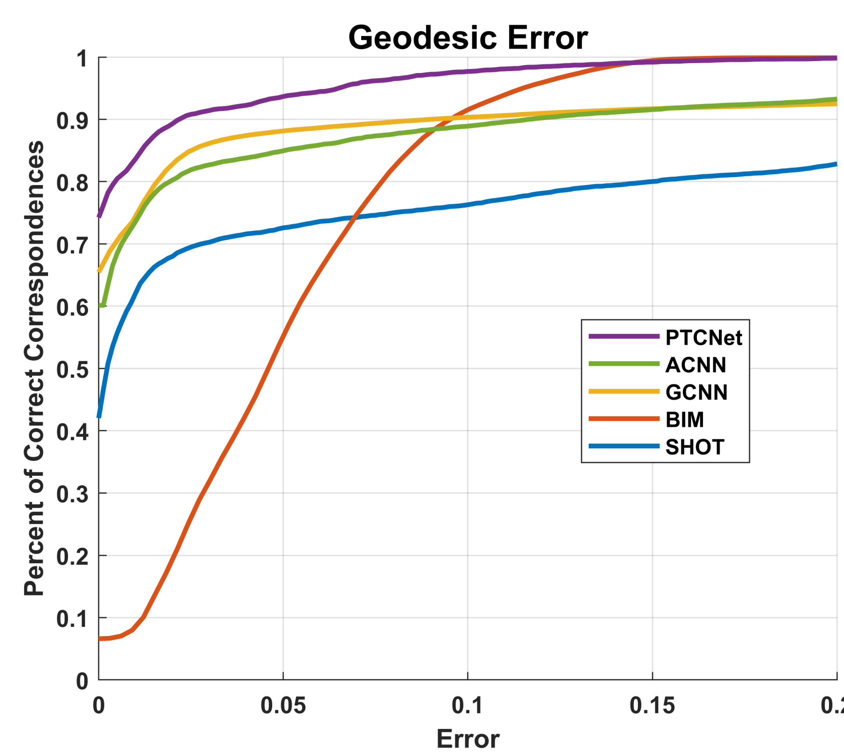
Manifold registration using PTCNN







Surface registration



A siamese neural network [Bromley94, Hadsell06, Masci15]

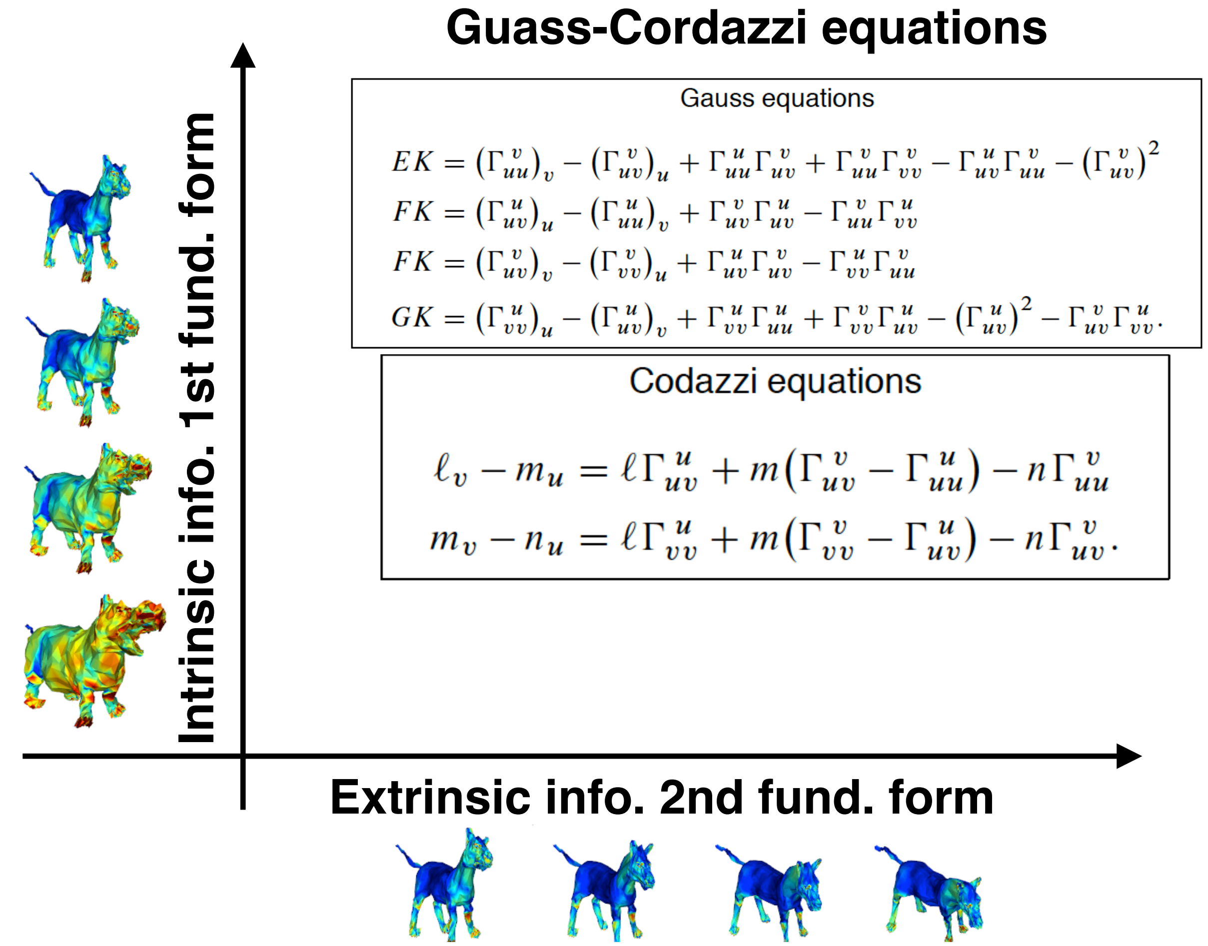
$$E(\Theta) = \frac{1}{2}\sum_{k=1}^{\ell}\left[\sum_{+}\|F_\Theta(\{\boldsymbol{f}_i^k\}) - F_\Theta(\{\boldsymbol{f}_i^{k,+}\})\|_{\mathcal{M}}^2 + \lambda\sum_{-}\max\{0, \mu - \|F_\Theta(\{\boldsymbol{f}_i^k\}) - F_\Theta(\{\boldsymbol{f}_i^{k,-}\})\|_{\mathcal{M}}\}^2\right]$$

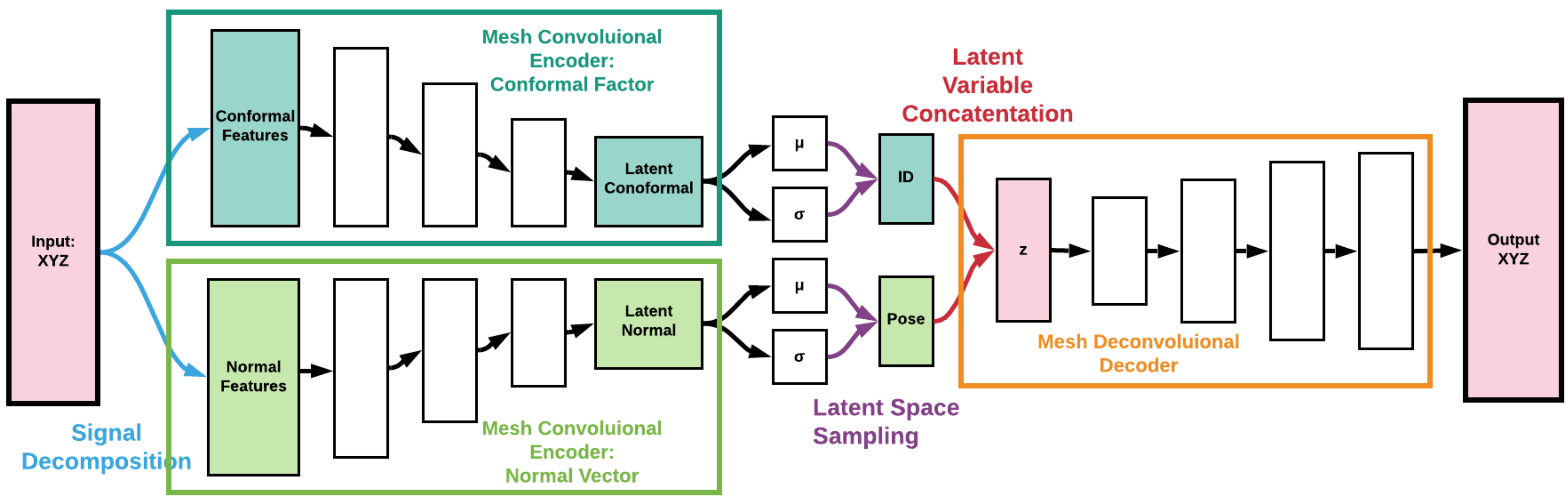
where ℓ is the number of training data set, $F_\Theta(\{f_i^{k,+}\})$ is the feature set on shapes similar to the k^{th} shape and $F_\Theta(\{f_i^{k,-}\})$ is the feature set on shape dissimilar to the k^{th} shape.

Unsupervised geometric disentanglement for Surfaces via CFAN-VAE (Tatro-Schonsheck-Lai'20)

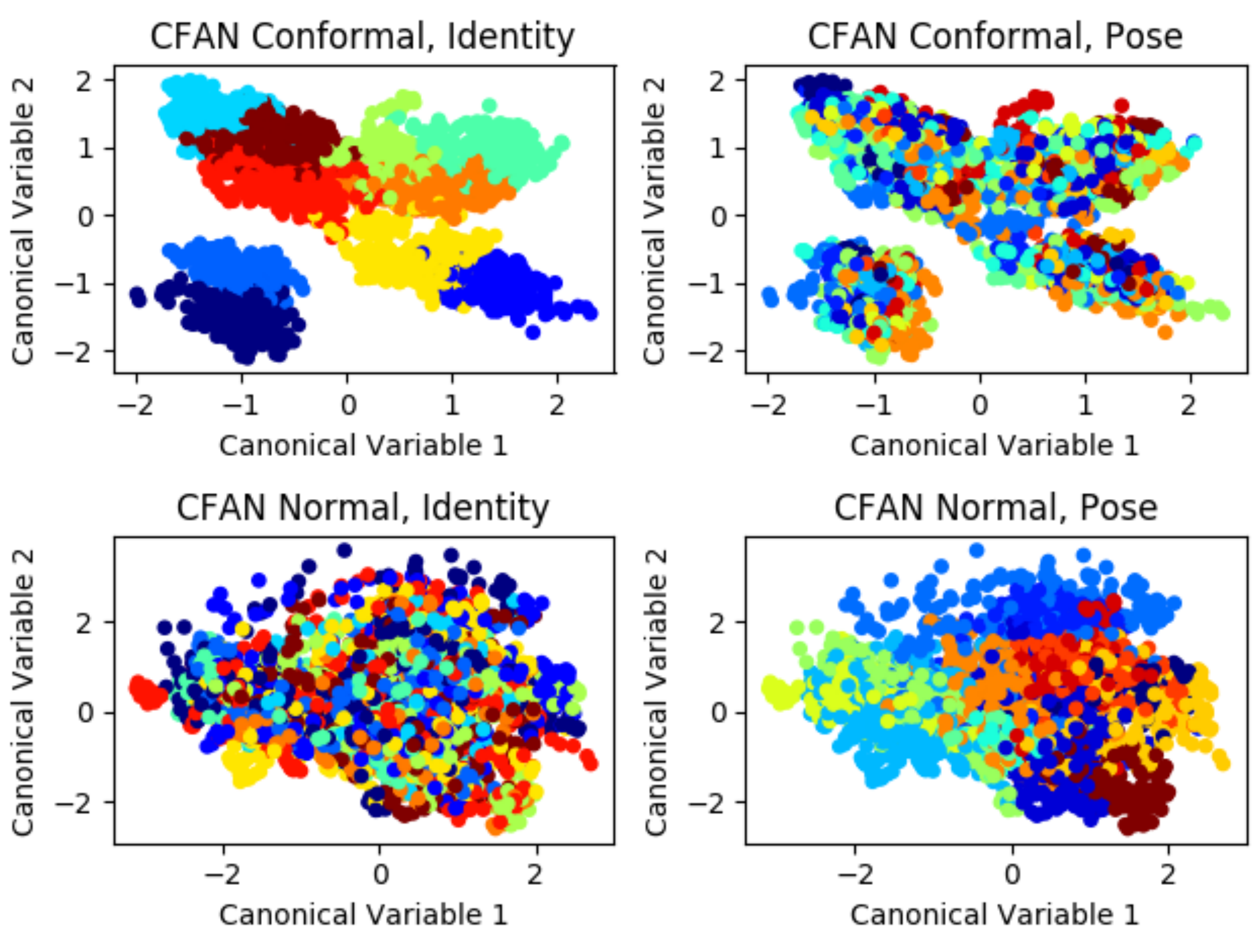
- We aim at design an unsupervised method to disentangle intrinsic and extrinsic information.
- Motivated by all genus-0 surfaces are conformally equivalent, we characterize surface using its conformal factor (1st fundamental form) and normal feature (2nd fundamental form) as:

$$c_i := \log\left(\sum_{\tau\in T; i\in\tau} \frac{\text{Area}(\tau)}{3}\right), \qquad \boldsymbol{n}_i := \frac{\sum_{\tau\in T; i\in\tau} \text{Area}(\tau)\boldsymbol{n}_\tau}{\|\sum_{\tau\in T; i\in\tau} \text{Area}(\tau)\boldsymbol{n}_\tau\|}$$

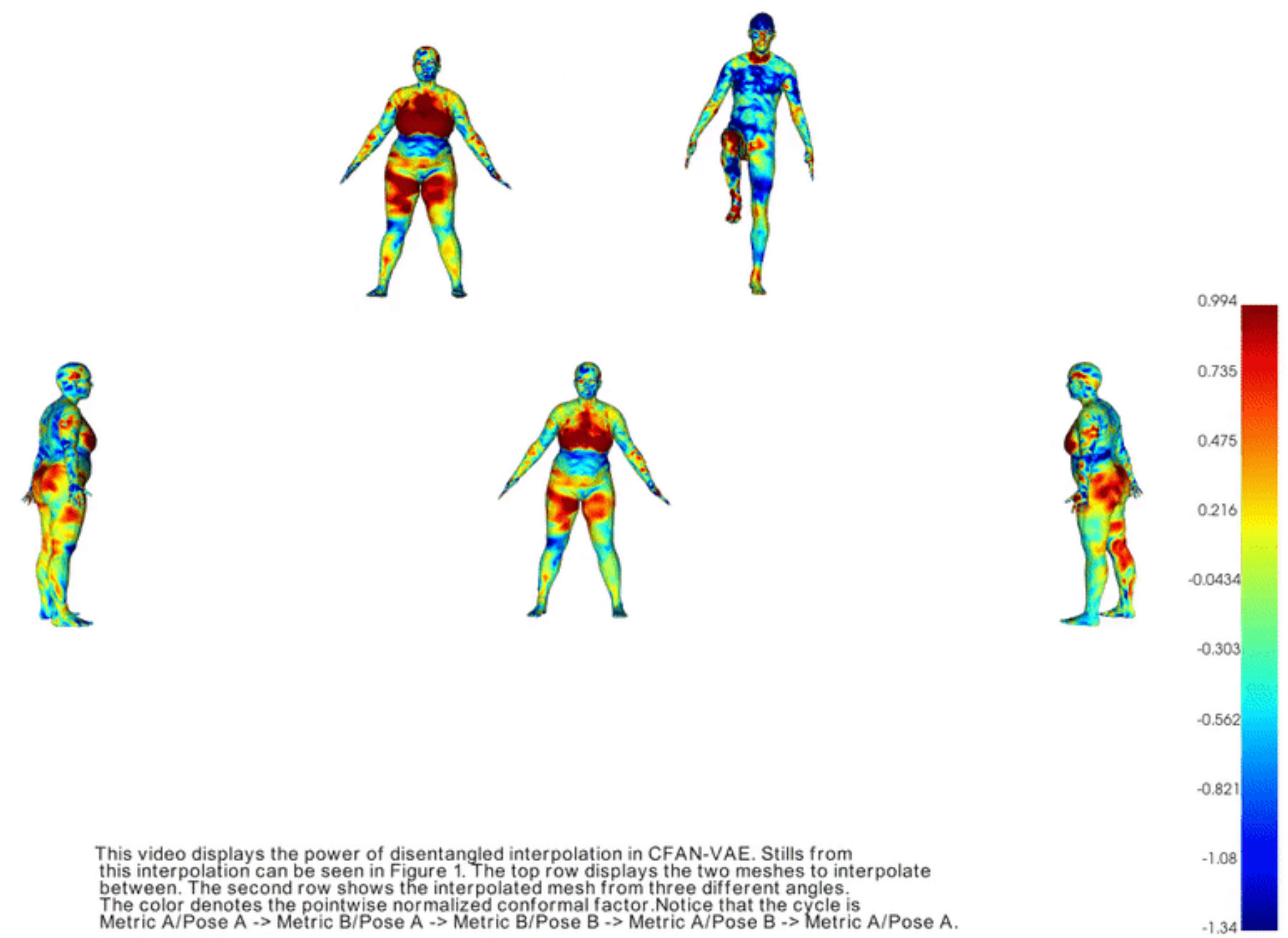




Geometric disentanglement and disentangled evolution paths



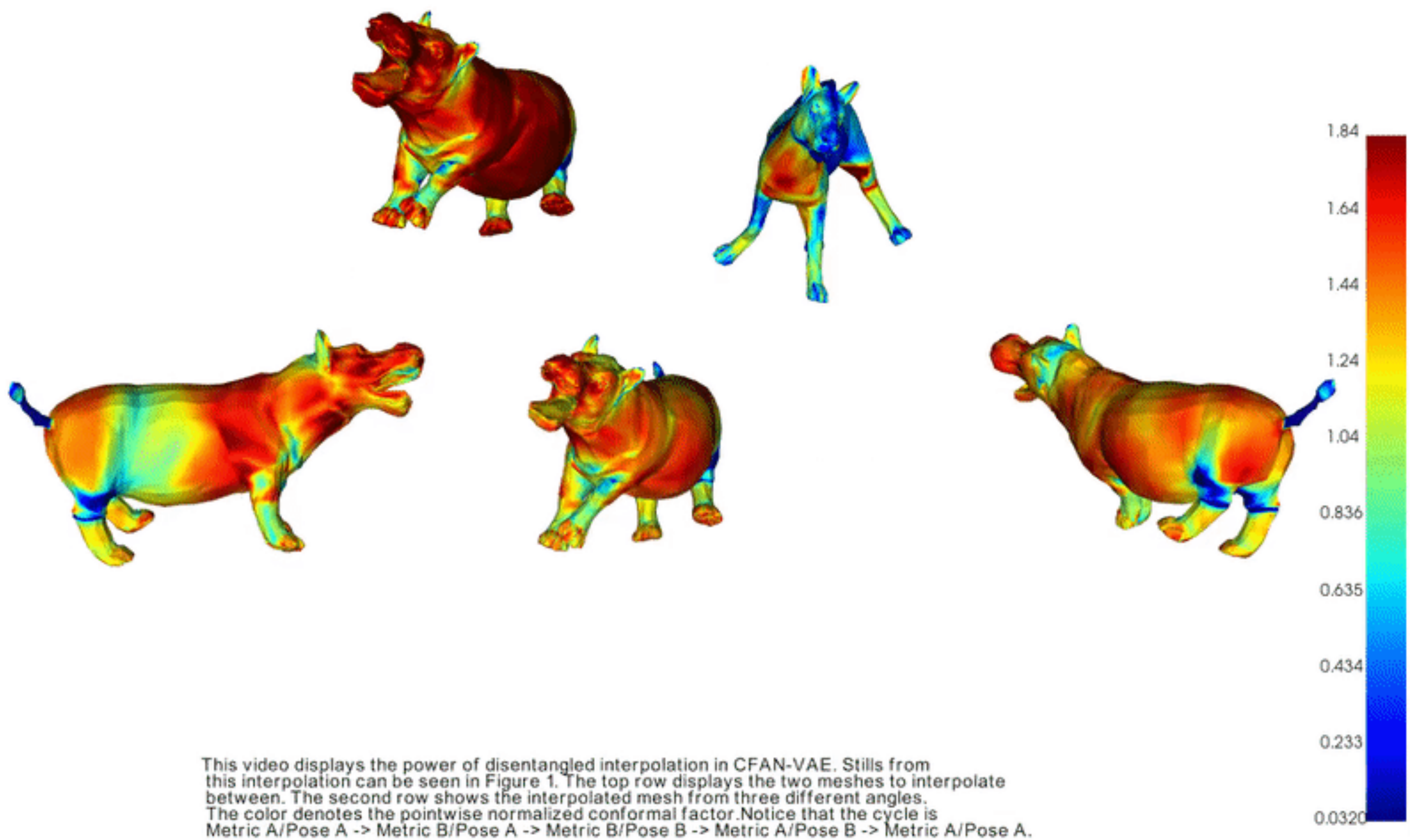
Source **Target**



DFAUST: Evolution paths of fixing pose and metric, respectively

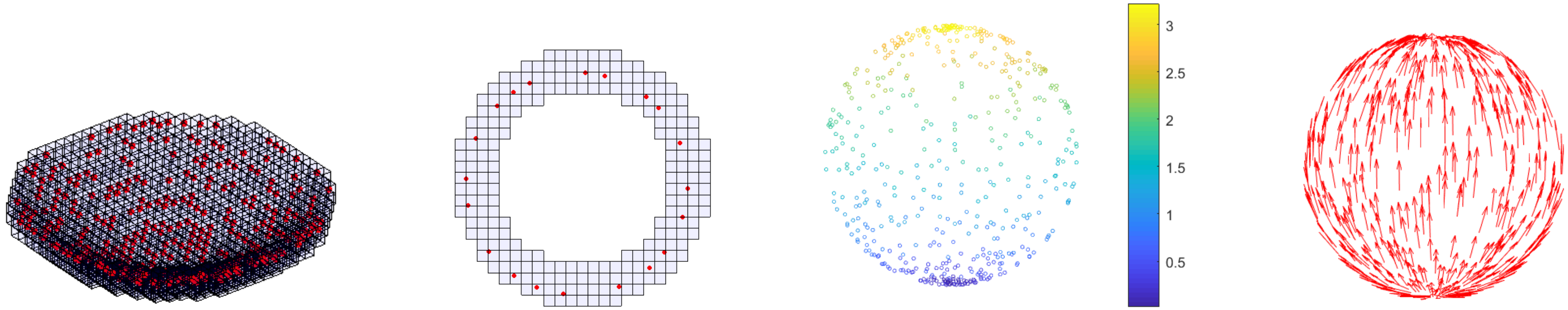
Geometric disentanglement and disentangled evolution paths

Source **Target**



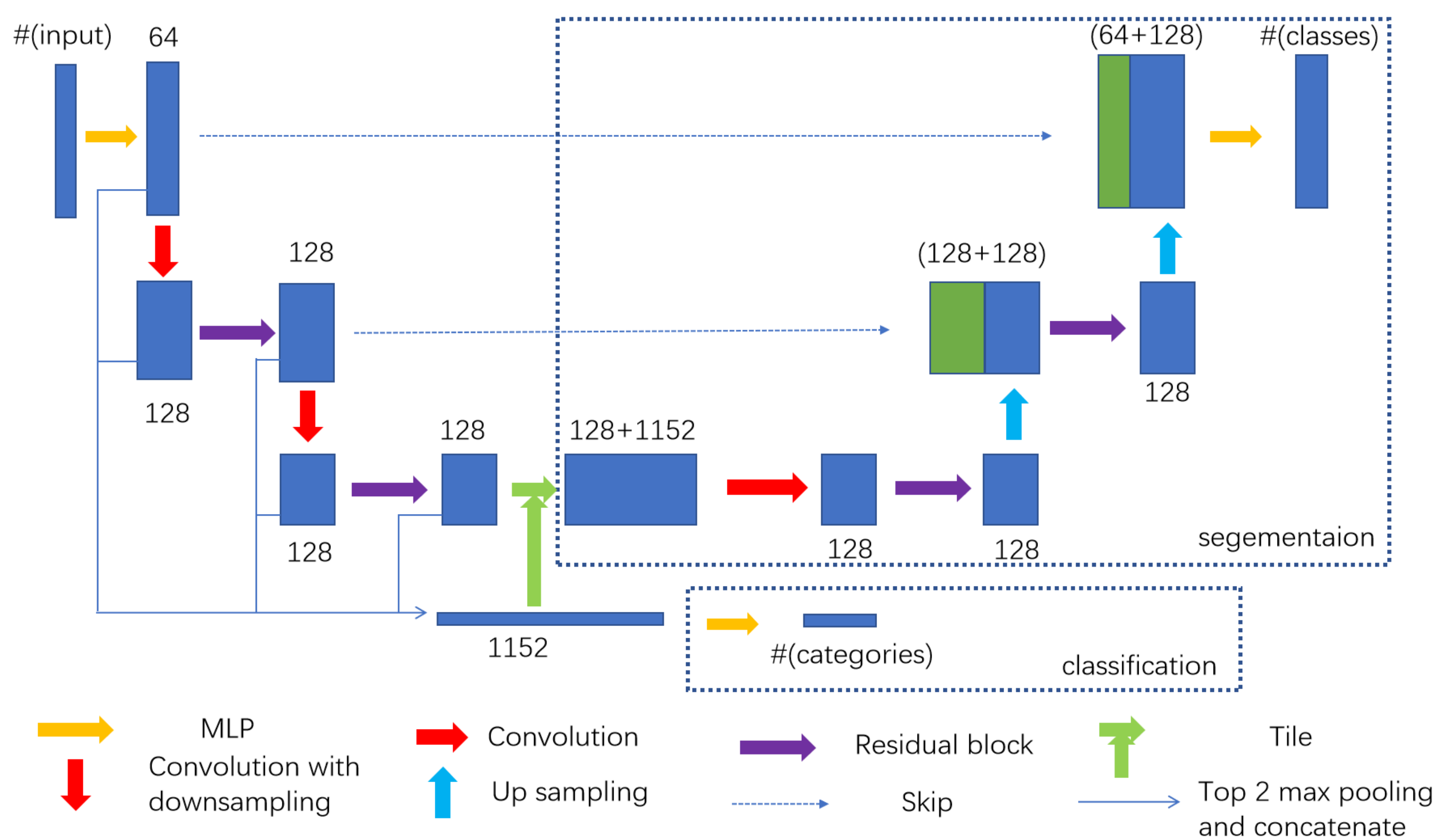
SMAL: Evolution paths of fixing pose and metric, respectively

Narrowband PTC on point clouds [Jin-Lai-Lai-Dong'22]



(a) narrow-band (b) cross section (c) Distance function ρ (d) $\{\vec{u}_x^1\}$

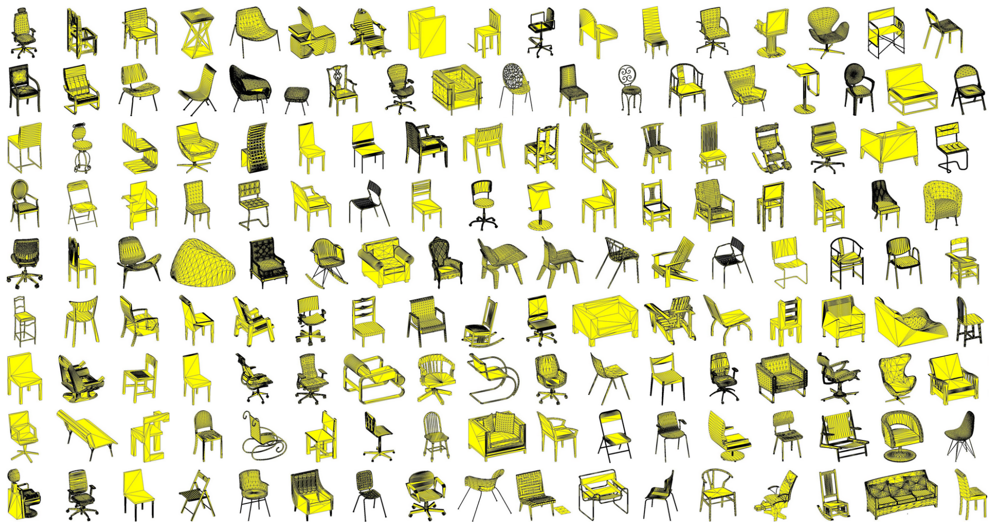
Figure 2: Illustration of a point cloud $\mathcal{P}$ sampled from the unit sphere. (a) shows the narrow-band approximation (blue boxes) of part of $\mathcal{P}$ (in red). (b) is a cross section of (a). (c), (d) show the distance function ρ and vector field $\{\vec{u}_x^1\}$ ($\{\nabla_{\mathcal{P}}\rho(x)\}$) on the point cloud. We can see that distance propagates from the bottom center to the top center reflecting the geometry of the sphere.

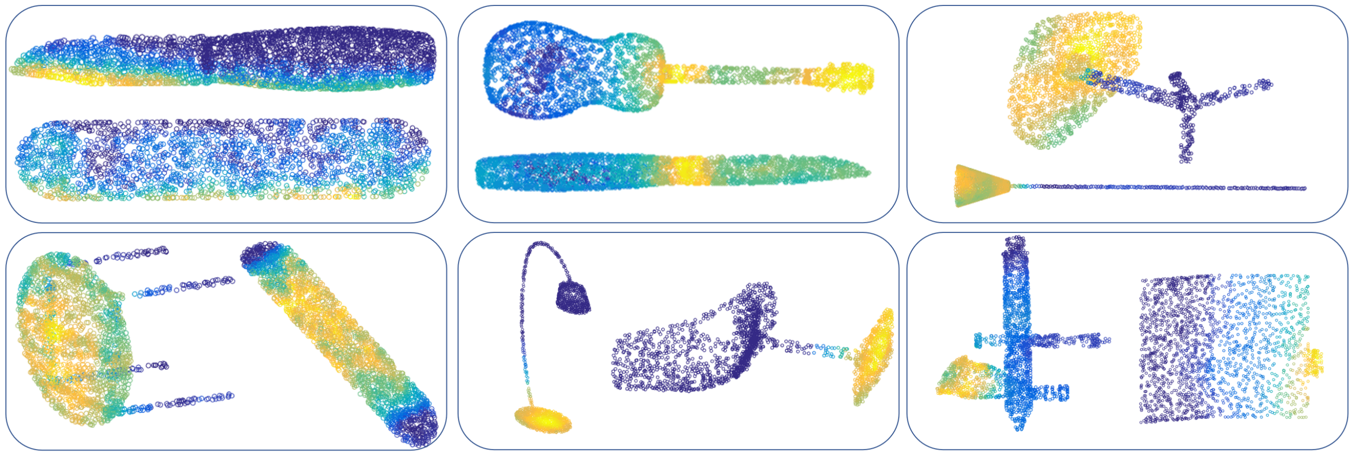


Classification and Segmentation [Narrowband PTC, Jin-Lai-Lai-Dong]

Table 1: Comparisons of overall accuracy (OA) and mean per-class accuracy (mA) on ModelNet40 as well as comparisons in instance average IoU (mIoU) and class average IoU (mcIoU) on ShapeNet Part. Models ranking first is colored in red and second in blue.

	Modelnet40		ShapeNet part	
Method	OA(%)	mA(%)	mIoU	mcIoU
kd-net Klokov & Lempitsky (2017)	91.8	88.5	82.3	77.4
pointnet Qi et al. (2017a)	89.2	86.2	83.7	80.4
SO-Net Li et al. (2018a)	90.9	87.3	84.9	81.0
pointnet++ Qi et al. (2017b)	90.7	-	85.1	81.9
SpecGCN Wang et al. (2018a)	92.1	-	85.4	-
SpiderCNN Xu et al. (2018)	92.4	-	85.3	81.7
pointcnn Li et al. (2018b)	92.2	88.1	86.1	84.6
DGCNN Wang et al. (2018c)	92.2	90.2	85.1	82.3
Ours	92.7	90.2	85.8	83.3





Robustness test [Narrowband PTC, Jin-Lai-Lai-Dong]

Table 2: Comparisons of overall accuracy (OA) and mean per-class IoU (mIoU) on S3DIS. Models ranking first is colored in red and second in blue.

Convolution Type	Method	OA(%)	mIoU(%)
no convolution	pointnet Qi et al. (2017a)	78.8	41.3
3-d convolution	SegCloud Tchapmi et al. (2017)	-	48.9
	Eff3DConv Zhang et al. (2018)	69.3	51.8
	ParamConv Wang et al. (2018b)	-	58.3
geometric convolution	TangentConv Xu et al. (2018)	82.5	52.8
	Ours	83.7	54.0

S3DIS covers 6 large-scale indoor areas from 3 different buildings for a total of 273 million points annotated with 13 classes. This is a real-word scanned dataset without normal and with noise.

Summary

- Inspired by differential geometry, we consider a multi-chart latent space to understand the geometric structure of latent space in generative models. we theoretically show structured latent space is necessary and provide approximation and generalization bound on training data size and network size. We also show CAE is robust to noise.

- We proposed a spatial way of defining convolution on manifolds using parallel transport which naturally incorporates geometry. This time domain definition enjoy flexibility to handle isotropic/anisotropic diffusion. We demonstrate its applications in shape matching, geometric disentanglement, point clouds classification and segmentation

Thanks for your attention!

Supported in part by an NSF CAREER Award (DMS-1752934), NSF DMS-2134168 and IBM AIRC

- **Stephan Schonscheck, Jie Chen, Rongjie Lai, Chart Auto-encoders for Manifold-Structured Data, 2020.**
- **N. Tatro, S. Schonsheck, R. Lai, Unsupervised Geometric Disentanglement for Surfaces via CFAN-VAE, ICLR workshop, 2021**
- **Pengfei Jin, Tianhao Lai, Rongjie Lai, Bin Dong, Narrow-band Parallel Transport Convolutional Neural Network on Point Clouds, JSC, 2022**
- **Stephan Schonsheck, Bin Dong, Rongjie Lai, Parallel Transport Convolution and applications to deep learning on Manifolds. SIIMS, 2022**
- **H. Liu, H. Havrilla, R. Lai, W.Liao, Deep Nonparametric Estimation of Intrinsic Data Structure by Chart Autoencoders: Generalization and Robustness, preprint, 2023**

Some details

2.2 Neural networks

In this paper, we consider feedforward neural networks (FNN) with the rectified linear unit $\mathrm{ReLU}(a) = \max\{a, 0\}$. An FNN with L layers is defined as

$$f(\mathbf{x}) = W_L \cdot \mathrm{ReLU}\left(W_{L-1} \cdots \mathrm{ReLU}(W_1\mathbf{x} + \mathbf{b}_1) + \cdots + \mathbf{b}_{L-1}\right) + \mathbf{b}_L, \tag{5}$$

where the W_i's are weight matrices, the $\mathbf{b}_i$'s are bias vectors, and ReLU is applied element-wisely. We define a class of neural networks with inputs in $\mathbb{R}^D$ and outputs in $\mathbb{R}^d$ as

$$\begin{aligned}\mathcal{F}(D, d; L, p, K, \kappa, R) = \{f : \mathbb{R}^D \to \mathbb{R}^d \mid\ & f \text{ has the form of (5) with } L \text{ layers and width bounded by } p,\\ & \|f\|_\infty \le R, \sum_{i=1}^{L} \|W_i\|_0 + \|\mathbf{b}_i\|_0 \le K,\\ & \|W_i\|_{\infty,\infty} \le \kappa, \|\mathbf{b}_i\|_\infty \le \kappa \text{ for } i = 1, ..., L\},\end{aligned}$$

where $\|H\|_{\infty,\infty} = \max_{i,j} |H_{ij}|$ for a matrix H and $\|\cdot\|_0$ denotes the number of non-zero elements of its argument.

Theorem 3. Consider Setting 2. Let $\widehat{\mathscr{E}}, \widehat{\mathscr{D}}$ be a global minimizer of (8) with the network classes $\mathcal{F}_{\mathrm{NN}}^{\mathscr{E}} = \mathcal{F}(D, C_{\mathcal{M}}(d+1); L_{\mathscr{E}}, p_{\mathscr{E}}, K_{\mathscr{E}}, \kappa_{\mathscr{E}}, R_{\mathscr{E}})$ and $\mathcal{F}_{\mathrm{NN}}^{\mathscr{D}} = \mathcal{F}(C_{\mathcal{M}}(d+1), D; L_{\mathscr{D}}, p_{\mathscr{D}}, K_{\mathscr{D}}, \kappa_{\mathscr{D}}, R_{\mathscr{D}})$ where $C_{\mathcal{M}} = O((d \log d)(4/\tau)^d)$,

$$\begin{aligned} L_{\mathscr{E}} = O(\log^2 n + \log D),\ p_{\mathscr{E}} = O(Dn^{\frac{d}{d+2}}),\ K_{\mathscr{E}} = O((D\log D)n^{\frac{d}{d+2}} \log^2 n),\\ \kappa_{\mathscr{E}} = O(n^{\frac{2}{d+2}}),\ R_{\mathscr{E}} = O(\tau), \end{aligned} \tag{19}$$

$$\begin{aligned} L_{\mathscr{D}} = O(\log^2 n + \log D),\ p_{\mathscr{D}} = O(Dn^{\frac{d}{d+2}}),\ K_{\mathscr{D}} = O(Dn^{\frac{d}{d+2}} \log^2 n + D\log D),\\ \kappa_{\mathscr{D}} = O(n^{\frac{1}{d+2}}),\ R_{\mathscr{D}} = B. \end{aligned} \tag{20}$$

We have

$$\mathbb{E}_{\mathcal{S}} \mathbb{E}_{\mathbf{x}\sim\gamma} \|\widehat{\mathscr{D}} \circ \widehat{\mathscr{E}}(\mathbf{x}) - \mathbf{v}\|_2^2 \le C(D^2 \log^3 D) n^{-\frac{2}{d+2}} \log^4 n + C_1\sigma^2 \tag{21}$$

for some constant C depending on $d, \tau, q, B, M, C_{\mathcal{M}}$ and the volume of $\mathcal{M}$, and C_1 depending on τ, q. The constant hidden in O depends on $d, \tau, q, B, M, C_{\mathcal{M}}$ and the volume of $\mathcal{M}$.