A Statistical Manifold Framework for Point Cloud Data


Yonghyeon Lee*
Seoul National University


Seungyeon Kim*
Seoul National University

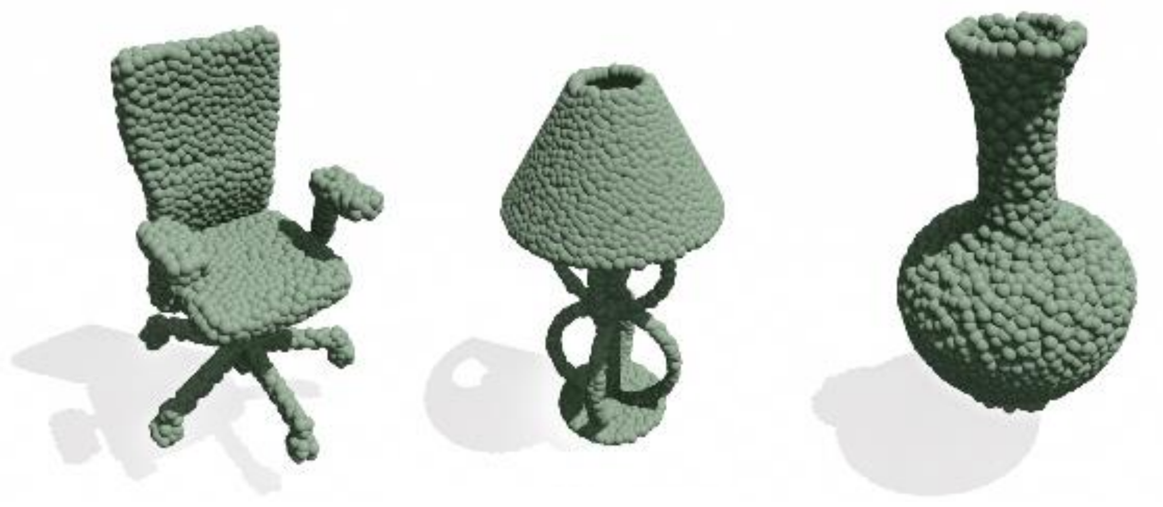

Jinwon Choi
Kakao Enterprise


Frank C. Park
Seoul National University
Saige Research

*: Equal contribution.

ICML 2022, 39th International Conference on Machine Learning

Point Cloud Data



3D point cloud data

Point Cloud Data

Point cloud data is a set

$$\mathbf{X} = \{x_1, x_2, \cdots, x_n | x_i \in \mathbb{R}^3\}$$

Point Cloud Distance Metric

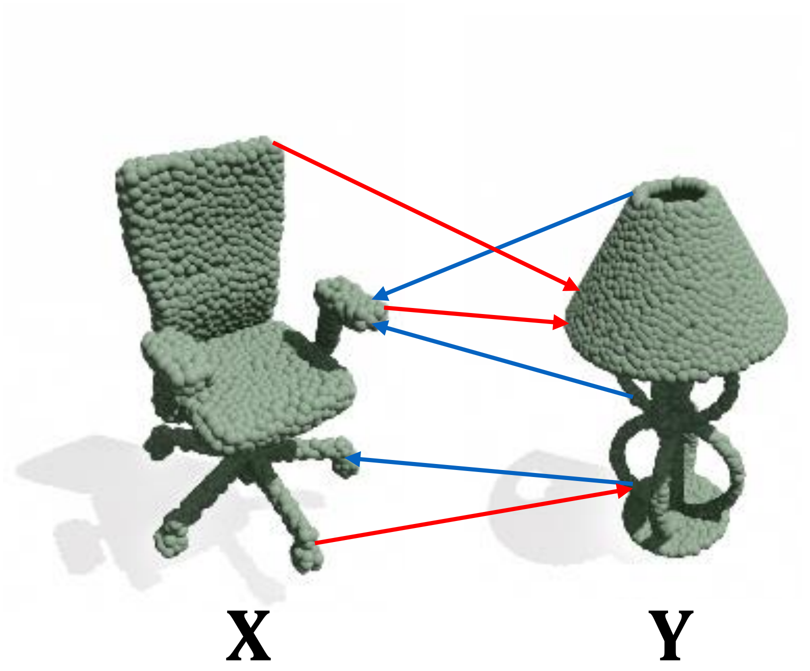
$\mathbf{X}$

$\mathbf{Y}$

Euclidean distance

$d(\mathbf{X}, \mathbf{Y}) = ||\mathbf{X} - \mathbf{Y}||$?

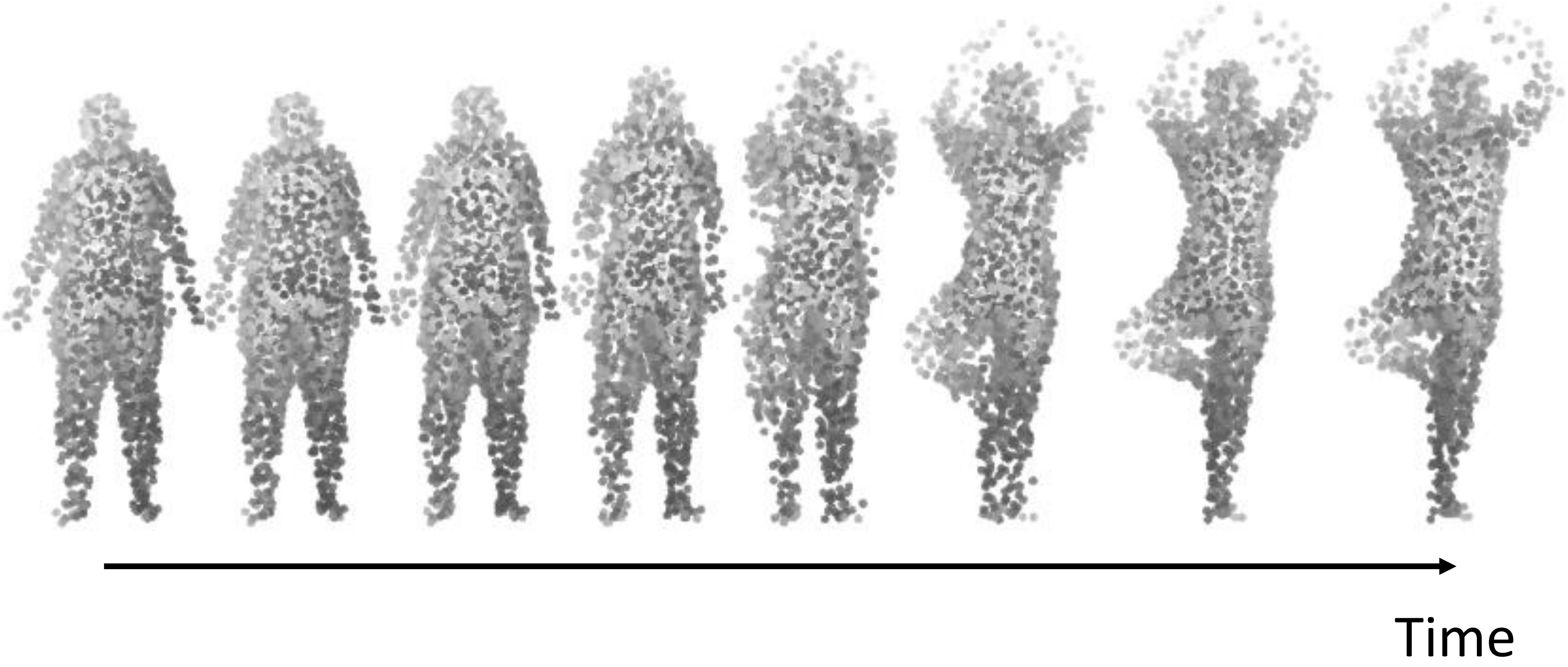
Point Cloud Distance Metric



Hausdorff distance

$$\max\left(\max_{x\in\mathbf{X}}\left(\min_{y\in\mathbf{Y}}||x-y||^2\right),\max_{y\in\mathbf{Y}}\left(\min_{x\in\mathbf{X}}||x-y||^2\right)\right)$$

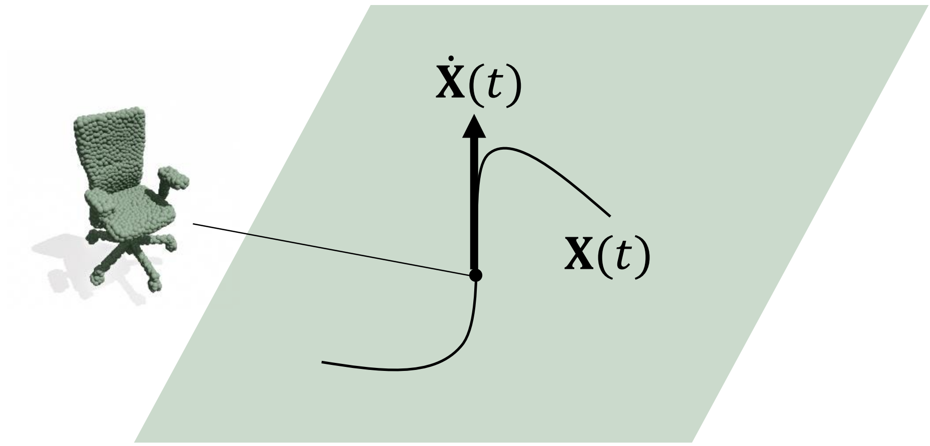
More Advanced Geometric Concepts



How to measure quantities such as the velocity and acceleration
of a moving point cloud?

The figure is excerpted from Cosmo, L., Norelli, A., Halimi, O., Kimmel, R., & Rodola, E. (2020, August). Limp: Learning latent shape representations with metric preservation priors. In *European Conference on Computer Vision* (pp. 19-35). Springer, Cham.

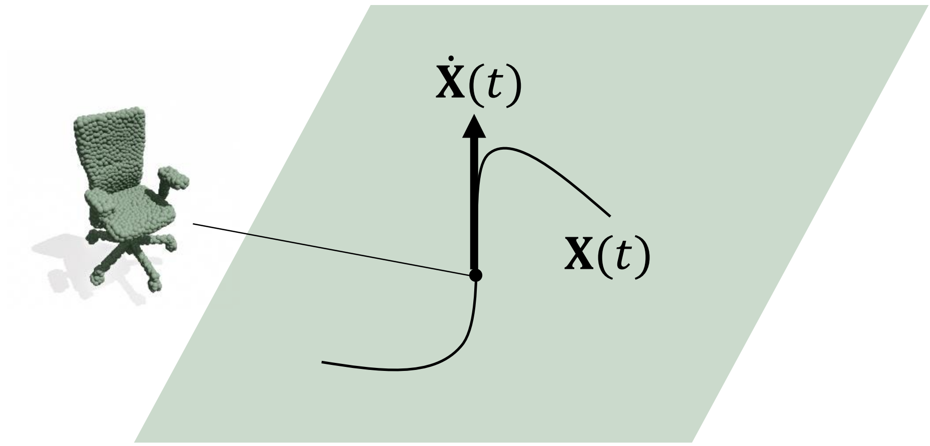
What about the usual “Euclidean” way?



- Euclidean norm of $\dot{\mathbf{X}} = \{\dot{x}_1, \dot{x}_2, \cdots, \dot{x}_n | \dot{x}_i \in \mathbb{R}^3\}$:

$$||\dot{\mathbf{X}}||^{\mathbf{2}} := \sum_{i=1}^{n} \dot{x}_i^T \dot{x}_i \,.$$

What about the usual “Euclidean” way?



- Euclidean norm of $\dot{\mathbf{X}} = \{\dot{x}_1, \dot{x}_2, \cdots, \dot{x}_n | \dot{x}_i \in \mathbb{R}^3\}$:

$$||\dot{\mathbf{X}}||^{\mathbf{2}} := \sum_{i=1}^{n} \dot{x}_i^T \dot{x}_i \,.$$

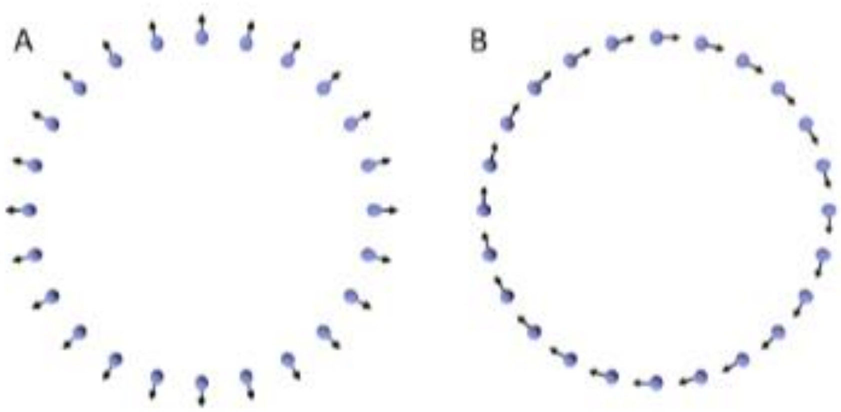
- For the following matrix representation

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times 3},$$

the Euclidean norm is written as

$$||\dot{\mathbf{X}}||^{\mathbf{2}} = \dot{X}^T \dot{X}.$$

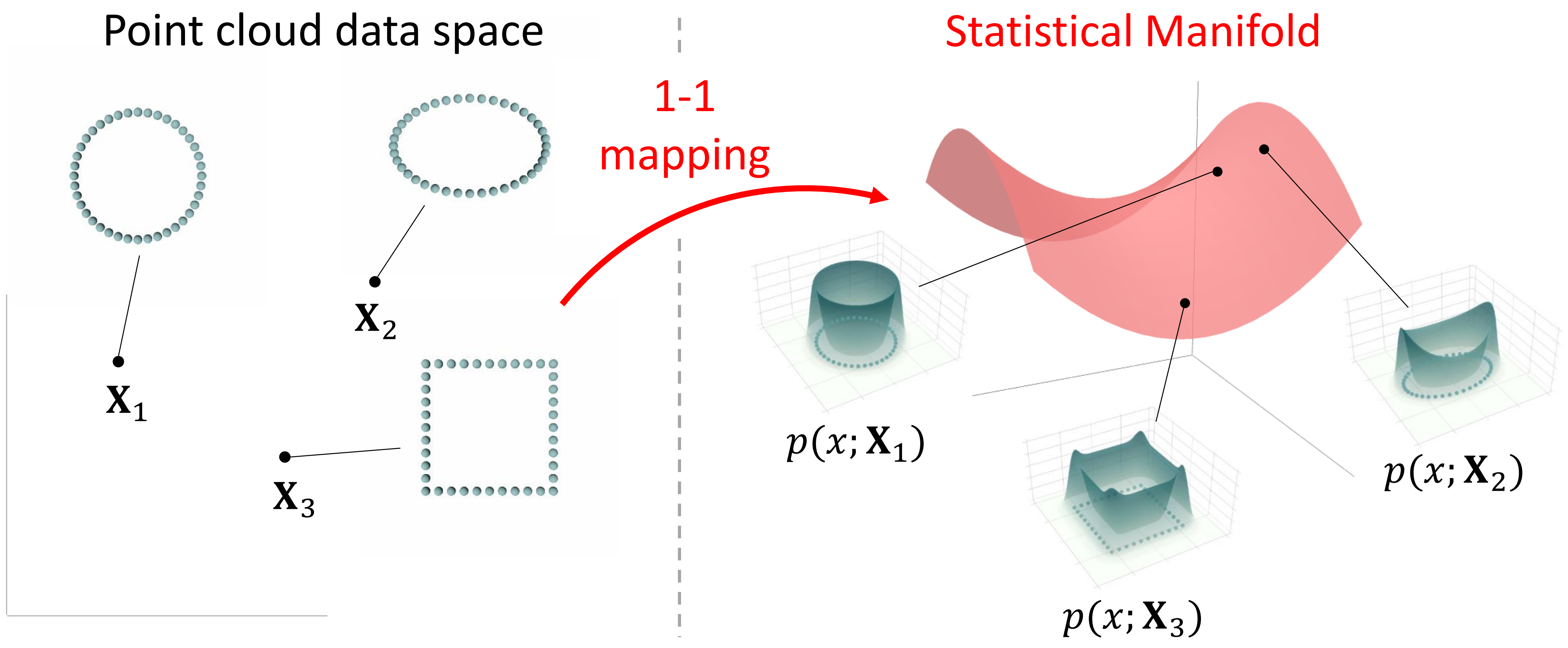
What about the usual “Euclidean” way?



Both velocities have the same Euclidean norm

➡ Euclidean norm does not capture the change in “shape”

Statistical Manifold Framework



Statistical Manifold Framework

With the Fisher information metric acting as a natural Riemannian metric* for the statistical manifold, the Riemannian norm can be defined as follows:

$$||\dot{\mathbf{X}}||^2 := \sum_{i,k=1}^{n} \sum_{j,l=1}^{3} \mathbb{E}_{x\sim p(x;\mathbf{X})} \left[\frac{\partial \log p(x;\mathbf{X})}{\partial X^{ij}} \frac{\partial \log p(x;\mathbf{X})}{\partial X^{kl}} \right] \dot{X}^{ij} \dot{X}^{kl} ,$$

where X is the matrix representation of the point cloud $\mathbf{X}$ such that $X = \left(X^{ab}\right)$ for $a = 1, \cdots, n$ and $b = 1,2,3$.

A Riemannian metric* allows one to define distances and angles near each point of a surface (or, more generally, a manifold), in the same way distances and angles in Euclidean space can be defined via an inner product.

Statistical Manifold Framework

With the Fisher information metric acting as a natural Riemannian metric* for the statistical manifold, the Riemannian norm can be defined as follows:

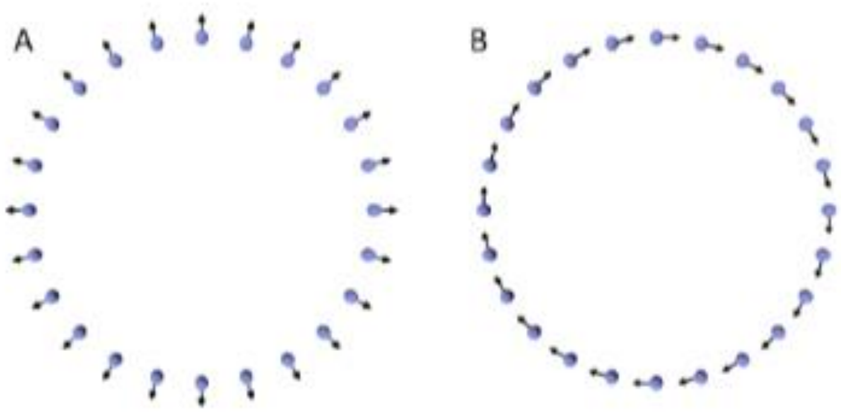
$$||\dot{\mathbf{X}}||^2 := \sum_{i,k=1}^{n} \sum_{j,l=1}^{3} \mathbb{E}_{x \sim p(x;\mathbf{X})} \left[\frac{\partial \log p(x;\mathbf{X})}{\partial X^{ij}} \frac{\partial \log p(x;\mathbf{X})}{\partial X^{kl}} \right] \dot{X}^{ij} \dot{X}^{kl},$$

where X is the matrix representation of the point cloud $\mathbf{X}$ such that $X = \left(X^{ab}\right)$ for $a = 1, \cdots, n$ and $b = 1,2,3$.

$$|| \frac{d}{dt} \log p(x;\mathbf{X}) ||^2$$

It captures the change in "distribution"

In the proposed "Riemannian" way

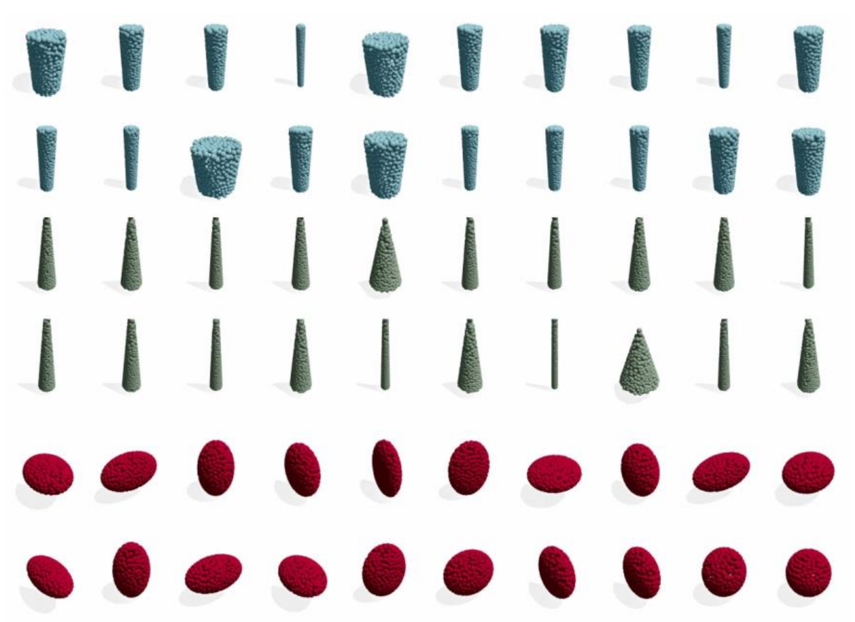


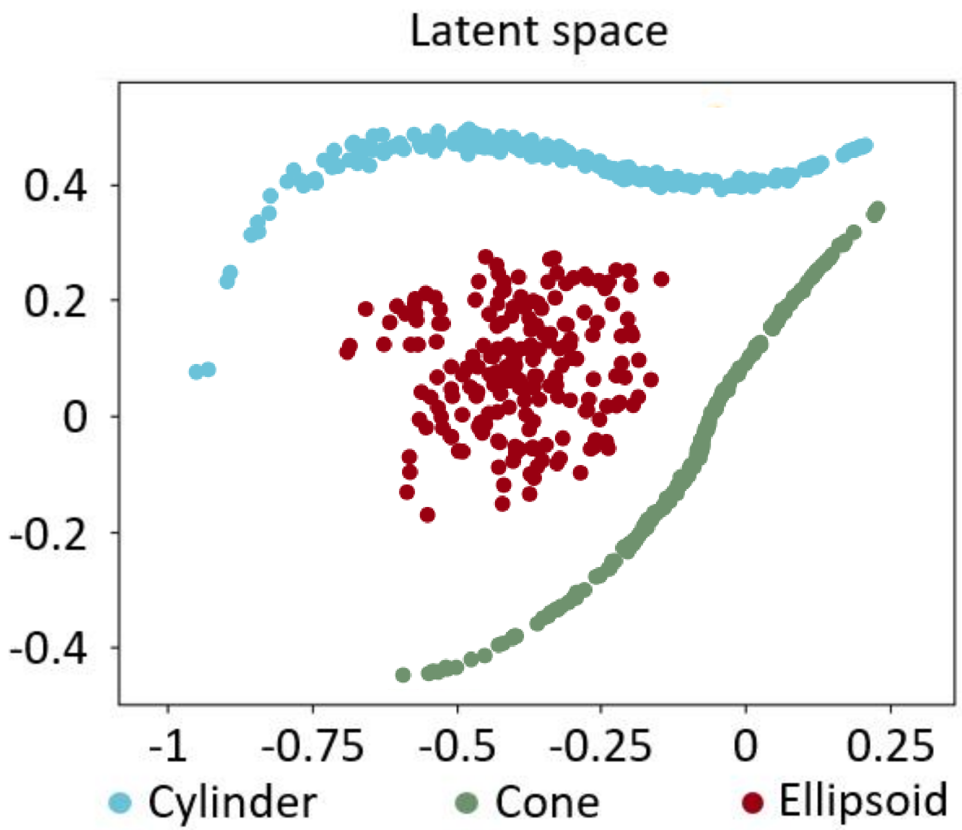
Both velocities have different Riemannian norms (larger for the case A than B)

➡ Riemannian norm can capture the change in "shape"

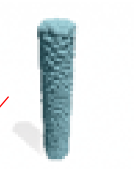
Autoencoder Applications

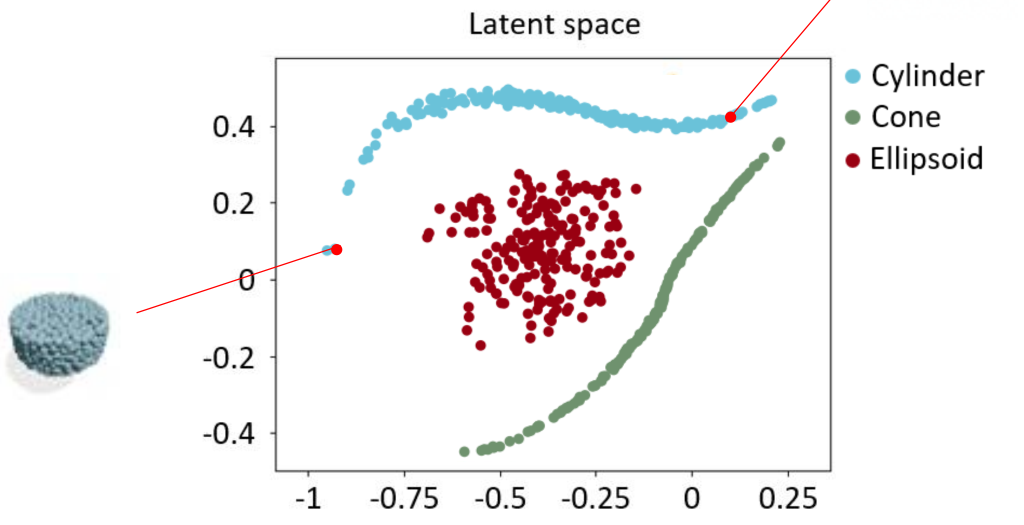
Training Data



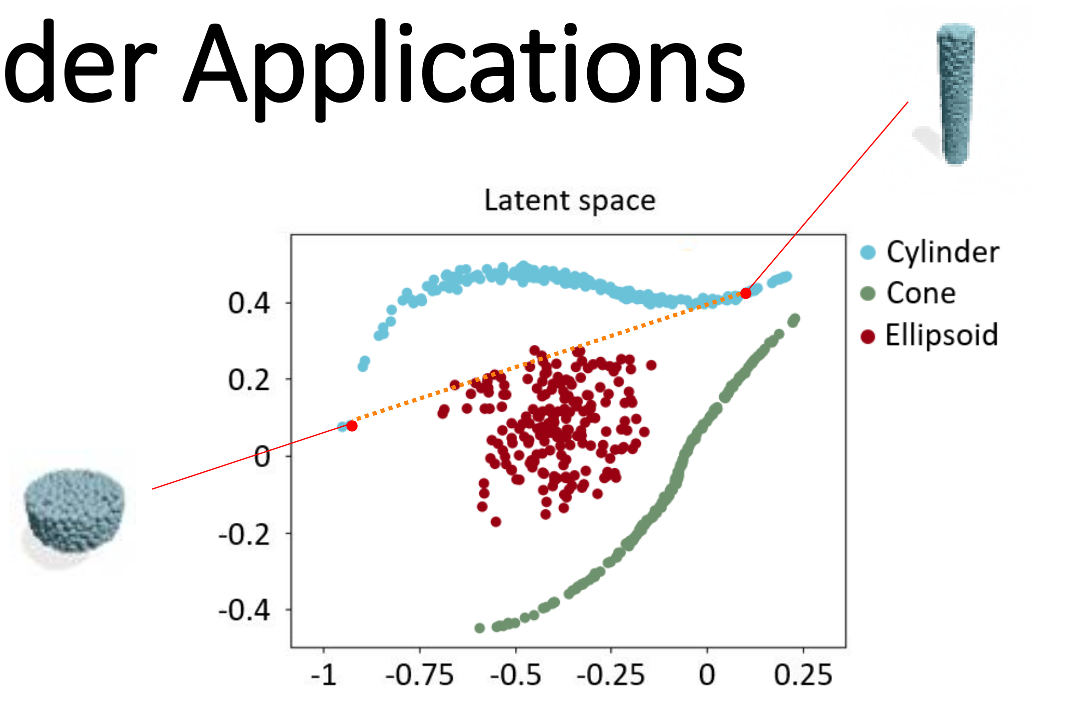


Autoencoder Applications



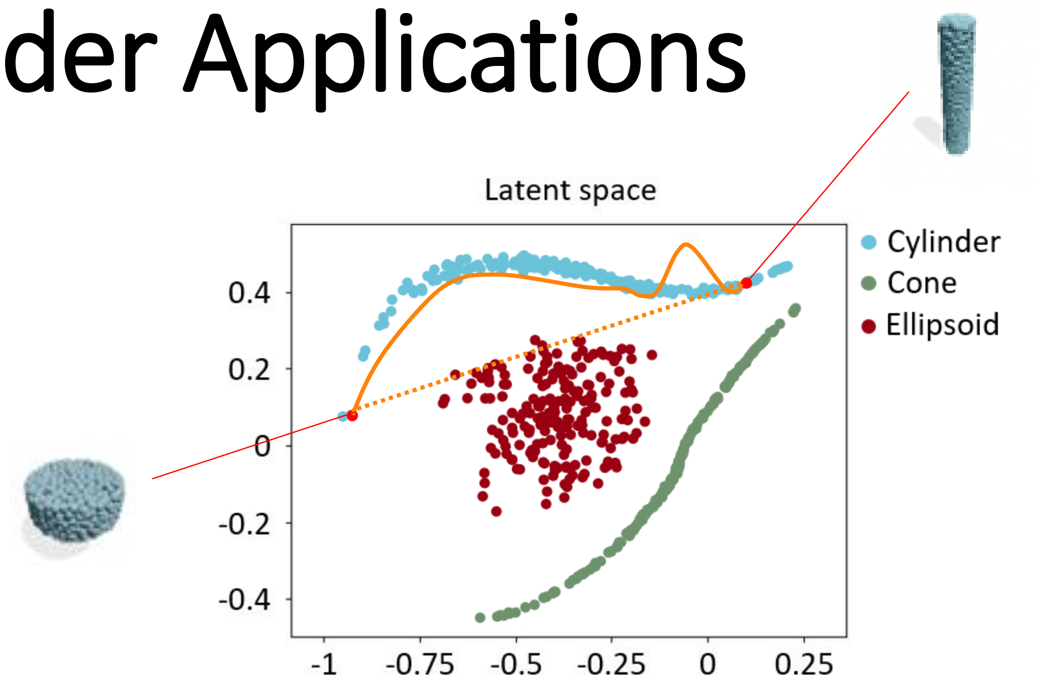


Autoencoder Applications



Linear interpolants

Autoencoder Applications

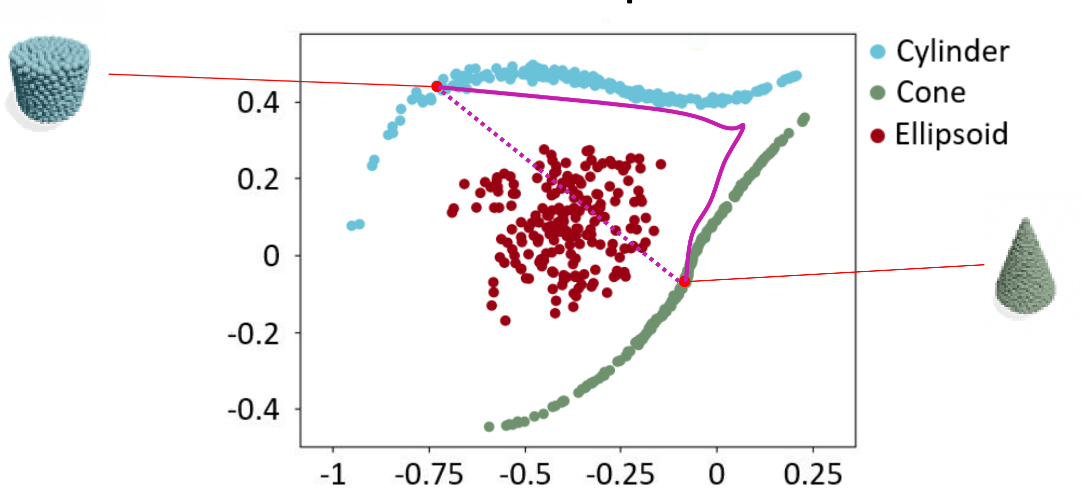


Linear interpolants

Riemannian geodesic interpolants

Autoencoder Applications

Latent space



Linear interpolants

Riemannian geodesic interpolants

Autoencoder Applications

Classification accuracy by transfer learning
for ModelNet10 (MN10) and ModelNet40 (MN40) from ShapeNet.

METHOD	MN40	MN10
FcNet	88.3%	93.5%
FcNet + E (ours)	89.3%	93.7%
FcNet + I (ours)	90.4%	94.3%
FoldingNet	89.3%	93.7%
FoldingNet + E (ours)	88.9%	94.4%
FoldingNet + I (ours)	90.1%	94.5%
PointCapsNet	87.2%	93.6%
PointCapsNet + E (ours)	88.1%	93.7%
PointCapsNet + I (ours)	88.5%	93.9%
DGCNN-FcNet	90.3%	94.5%
DGCNN-FcNet + E (ours)	89.9%	94.4%
DGCNN-FcNet + I (ours)	**91.0%**	**95.2%**

Vanilla setting

	MN40			
METHOD	1%	5%	10%	20%
FcNet	87.8%	83.2%	75.6%	64.5%
FcNet + E (ours)	86.6%	85.1%	79.1%	70.4%
FcNet + I (ours)	**89.0%**	**86.6%**	**81.4%**	**72.4%**
	MN10			
	1%	5%	10%	20%
FcNet	92.4%	91.9%	88.4%	79.8%
FcNet + E (ours)	92.2%	91.1%	88.2%	82.6%
FcNet + I (ours)	**93.3%**	**92.6%**	**91.6%**	**84.8%**

Noisy data setting

METHOD	MN40			
	50%	10%	5%	1%
FcNet	85.7%	78.0%	70.6%	50.3%
FcNet + I (ours)	**87.9%**	**81.6%**	**76.8%**	**57.4%**
	MN10			
	50%	10%	5%	1%
FcNet	91.7%	90.1%	87.2%	74.1%
FcNet + I (ours)	**93.2%**	**91.2%**	**88.3%**	**78.1%**

Semi-supervised learning setting

More in the Paper

- Formal and general descriptions of
  - ✓ point cloud statistical manifold
  - ✓ information Riemannian metric
- More experimental results
- Thorough comparison with recent related works

…and much more!

Thank you for listening!

Contact: yhlee@robotics.snu.ac.kr, sykim@robotics.snu.ac.kr
Code: https://github.com/seungyeon-k/SMF-public