


FoldingNet Point Cloud Autoencoder

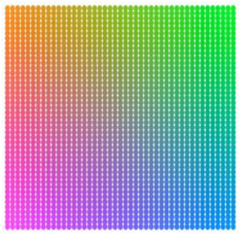
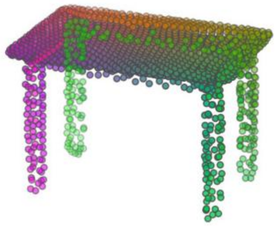
- Can Neural Networks Learn Paper Folding?

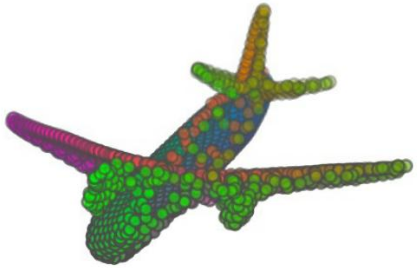
Yaoqing Yang

Carniegie Mellon University

(work done at MERL)

Thursday, Jan 31, 2019



The Papers and Collaborators

- FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation (CVPR’18 Spotlight)
- Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling (CVPR’18)
- Thank my collaborators for their support (including these slides)!

Dr. Chen Feng
NYU

Dr. Yiru Shen
Facebook

Dr. Dong Tian
InterDigital

This’s me!

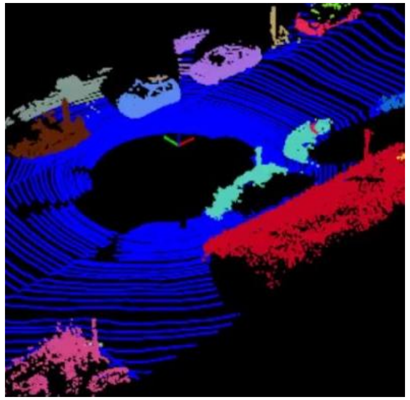
Code available:
http://www.merl.com/research/license#FoldingNet
http://www.merl.com/research/license#KCNet

Videos of the slides available:
https://www.youtube.com/watch?v=x1dAV4tP2oo

Deep Learning on 3D Data

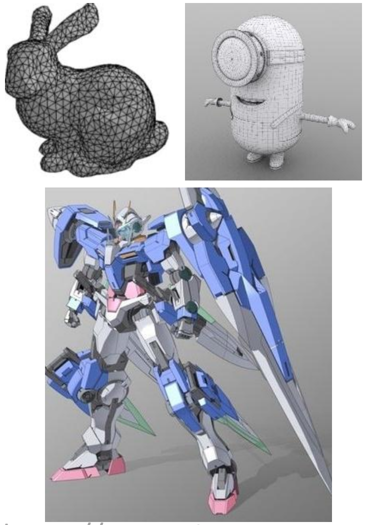
- Why 3D Deep Learning
  - Intrinsically different than images – E.g. unorganized/unordered
  - An important data format – many application domains

Robotics



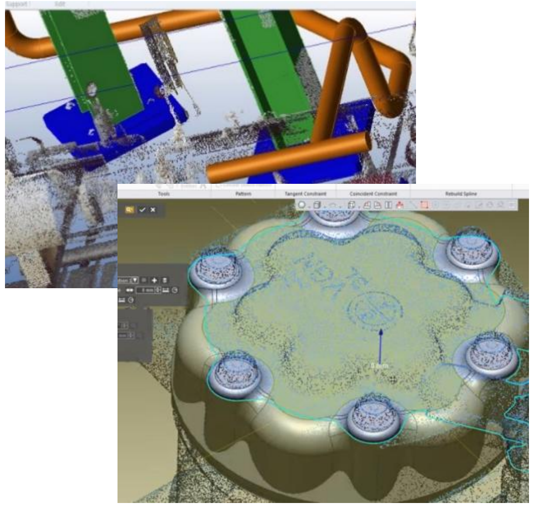
https://www.youtube.com/
watch?v=7NNpvtdrHkU

Graphics/3DP



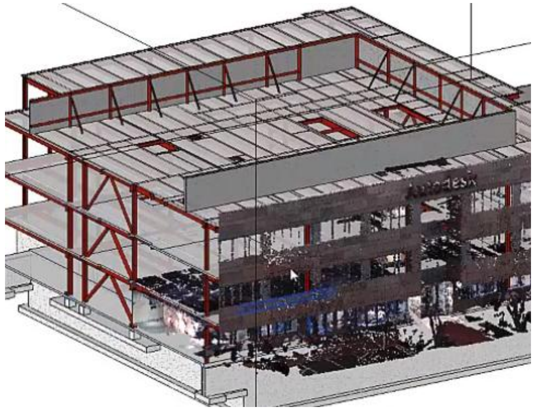
https://www.pinterest.com/
pin/134756213823244639/

Mechanical Engineering



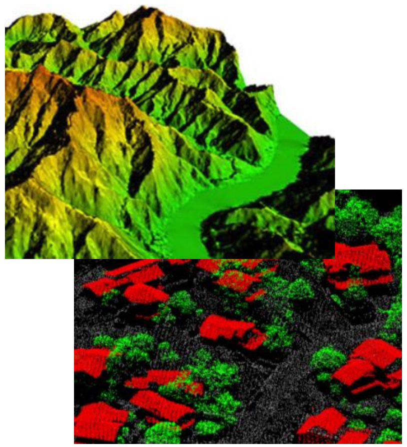
https://www.youtube.com/
watch?v=UD4asn3gkNI

Civil Engineering



https://www.youtube.com/
watch?v=HhV6LAZ3DN0

Geospatial Science



http://www.aamgroup.com/
services-and-technology/aerial-survey

3D Input Representation

Voxel

- ✓ 3D CNN
- • Implicit representation
- × Resolution/Scalability

https://www.planetminecraft.com/project/giant-snowman-1638162/

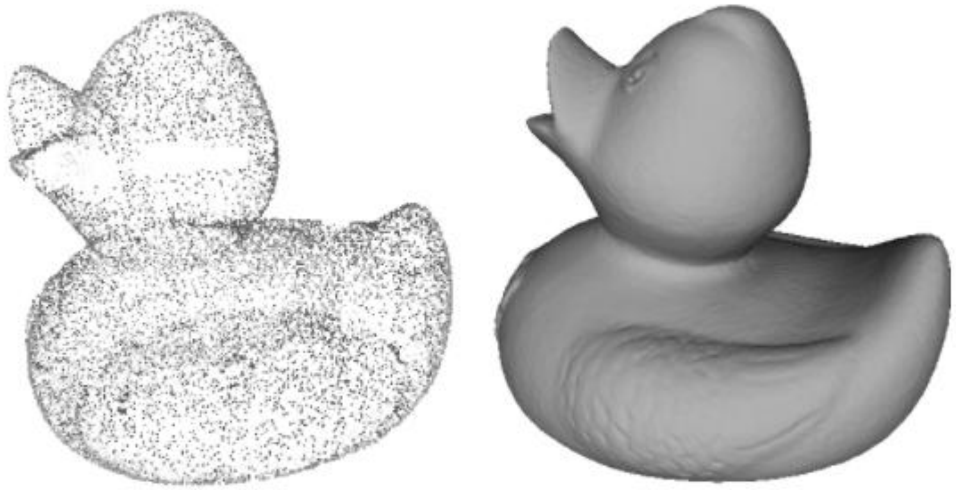
Multi-view

- ✓ 2D CNN
- • Generalize to points?
- × Large networks

http://photoboothexpo.com/bullet-time-photo-booths/

Point Cloud/Mesh

- ✓ Raw format/Efficiency
- • Explicit representation
- × Unorganized/Unordered



https://elmoatazbill.users.greyc.fr/point_cloud/reconstruction.png

FoldingNet

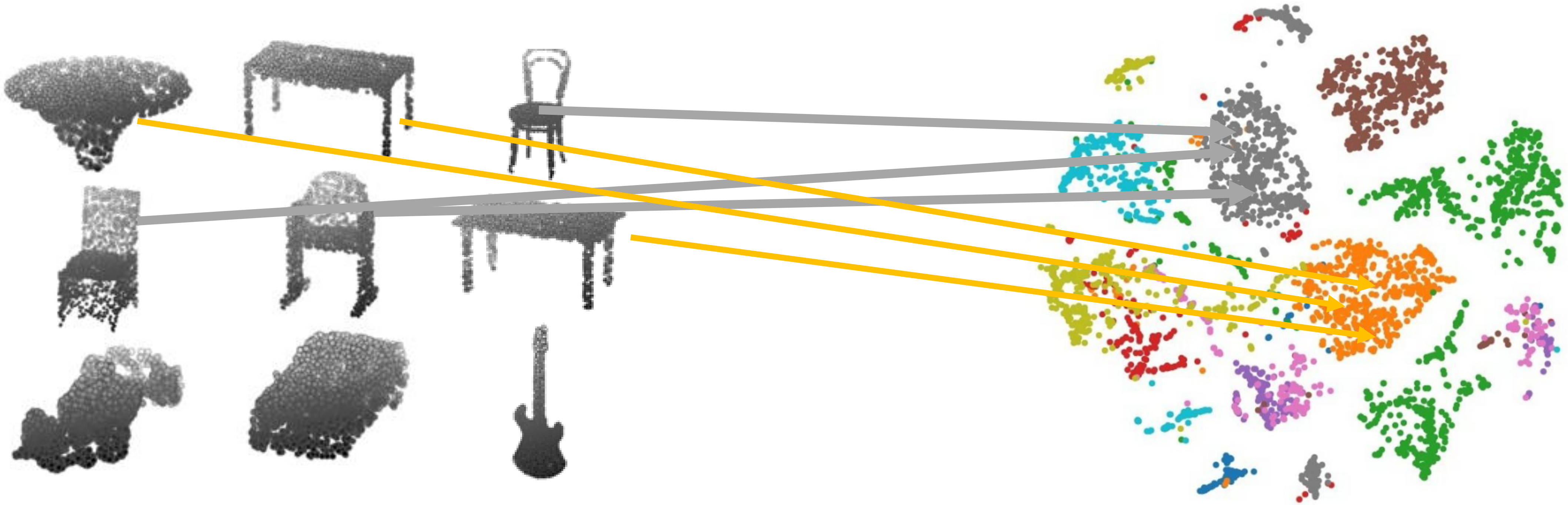
- Related works
- Conventional AutoEncoder
- Intuition – Paper Folding Operations
- FoldingNet Decoder Diagram
- Learned Folding Profiles
- A Theorem

Yang, Yaoqing, Chen Feng, Yiru Shen, and Dong Tian. "Foldingnet: Point cloud auto-encoder via deep grid deformation." In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 3. 2018.

What are we trying to do?

3D Data (Point Clouds)

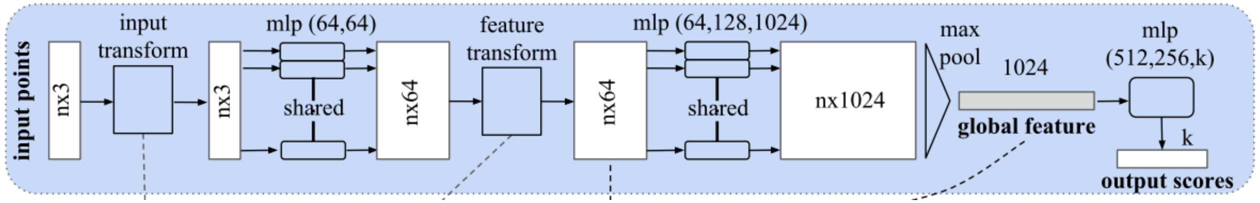
Latent space



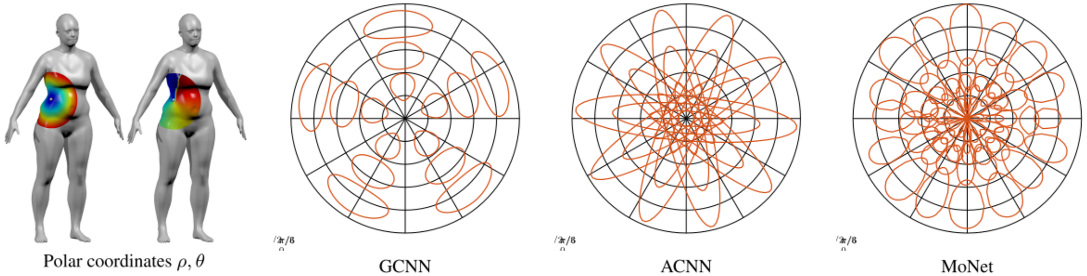
Unsupervised learning: reducing label cost, generation

Related Works: Deep Learning on Points

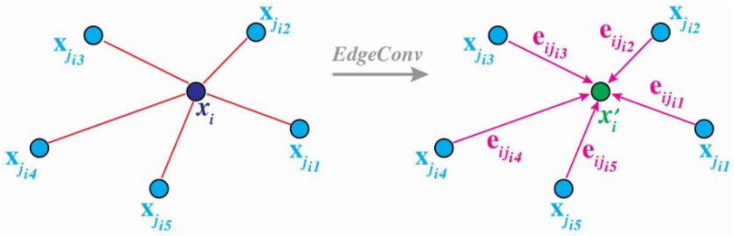
- PointNet [Qi et al., CVPR'17]
  - Share-weight MLP + Global Pooling
- MoNet [Monti et al., CVPR'17]
  - Graph/manifold/mesh
- Edge-Cond Graph CNN [Simonovsky et al., CVPR'17]
  Dynamic Graph CNN [Wang et al., ArXiv'18]
  - Edge feature function for Conv.
- And many new methods in CVPR'18!
  - SPLATNet, SO-Net, etc.



[Qi et al., CVPR'17]



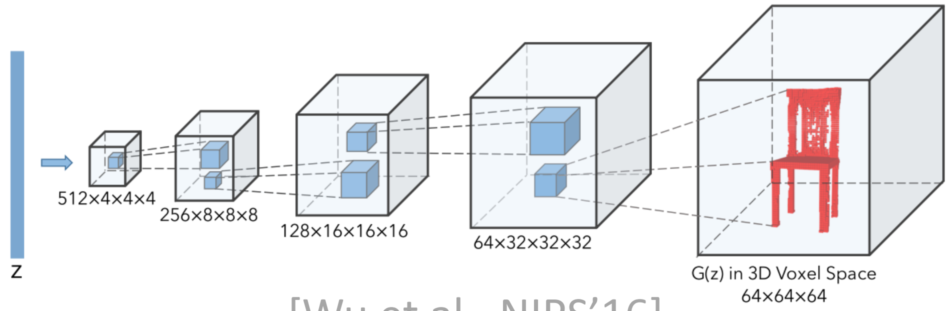
[Monti et al., CVPR'17]



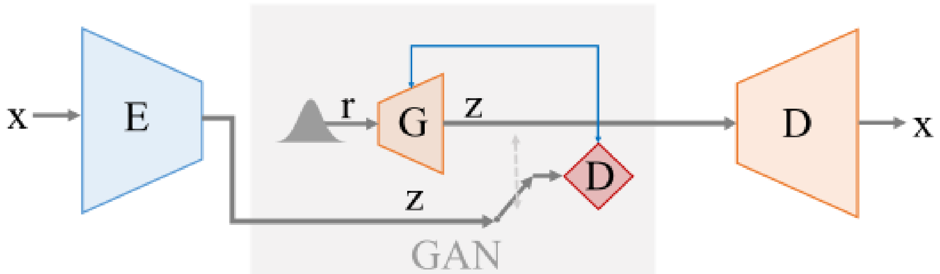
[Wang et al., ArXiv'18]

Related Works: Unsupervised 3D Deep Learning

- 3D-GAN [Wu et al., NIPS'16]
  - Voxel-based
  - Deconvolution-based decoder
- Latent-GAN [Achlioptas et al., arxiv'17]
  - Sort 3D points by lexicographic order
  - 1D CNN encoder
  - 3-fully-connected-layer decoder
- Point Set Generation Net [Fan et al., CVPR'17]
  - Supervised single image to point set
  - Deconvolution-based decoder



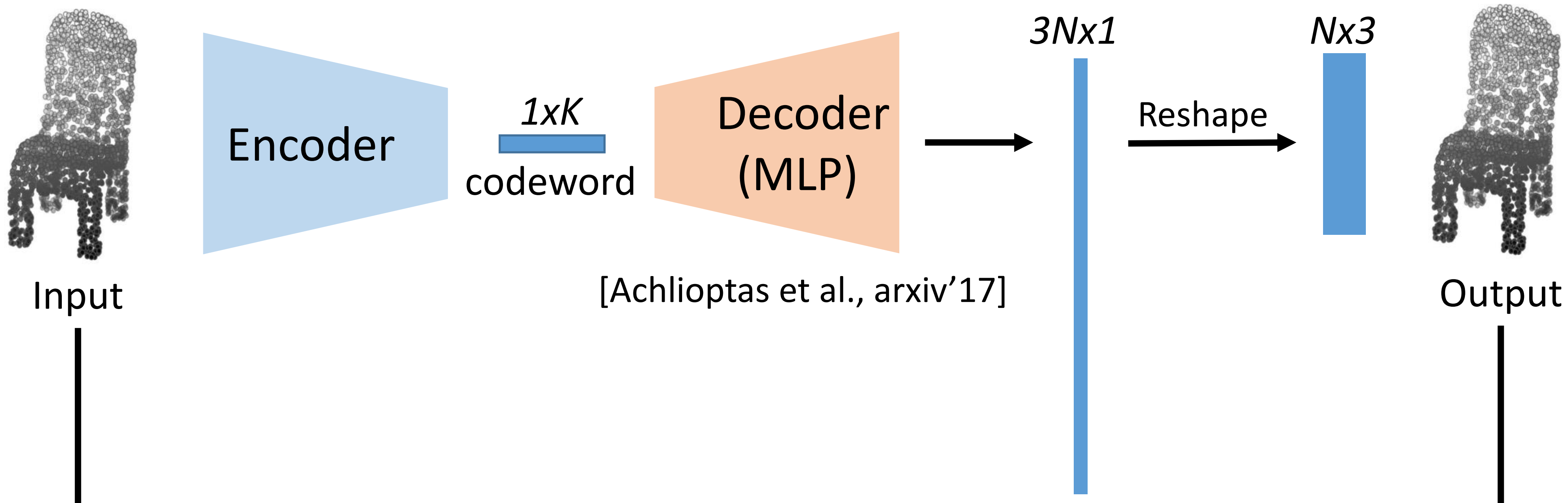
[Wu et al., NIPS'16]



[Achlioptas et al., arxiv'17]


[Fan et al., CVPR'17]

Baseline Auto-encoder Framework



Chamfer Distance (or EMD, etc.)

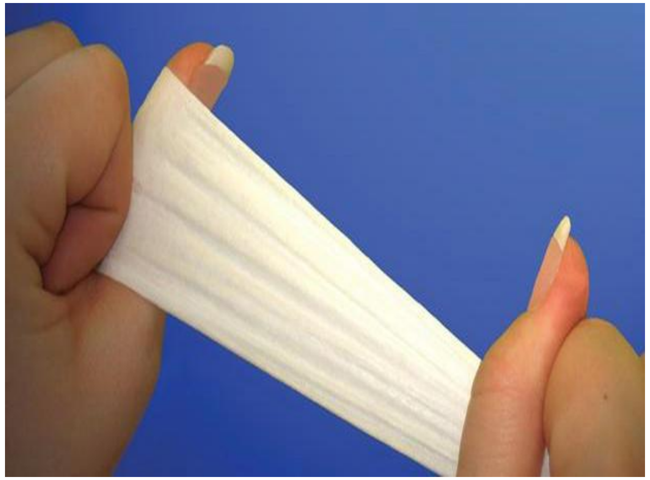
$$\max\left\{\frac{1}{|S|}\sum_{\mathbf{x}\in S}\min_{\hat{\mathbf{x}}\in\hat{S}}\|\mathbf{x}-\hat{\mathbf{x}}\|_2,\ \frac{1}{|\hat{S}|}\sum_{\hat{\mathbf{x}}\in\hat{S}}\min_{\mathbf{x}\in S}\|\hat{\mathbf{x}}-\mathbf{x}\|_2\right\}$$

Intuition of FoldingNet: Elastic Paper Folding

- 3D point clouds are often obtained from object surfaces
  - Discretized from CAD models
  - Sampled from line-of-sight sensors
- 3D object surfaces are intrinsically 2D-manifolds
  - Can be transformed from a 2D plane, through the Origami operations
  - This 2D-3D mapping is known as parameterization/cross-parameterization

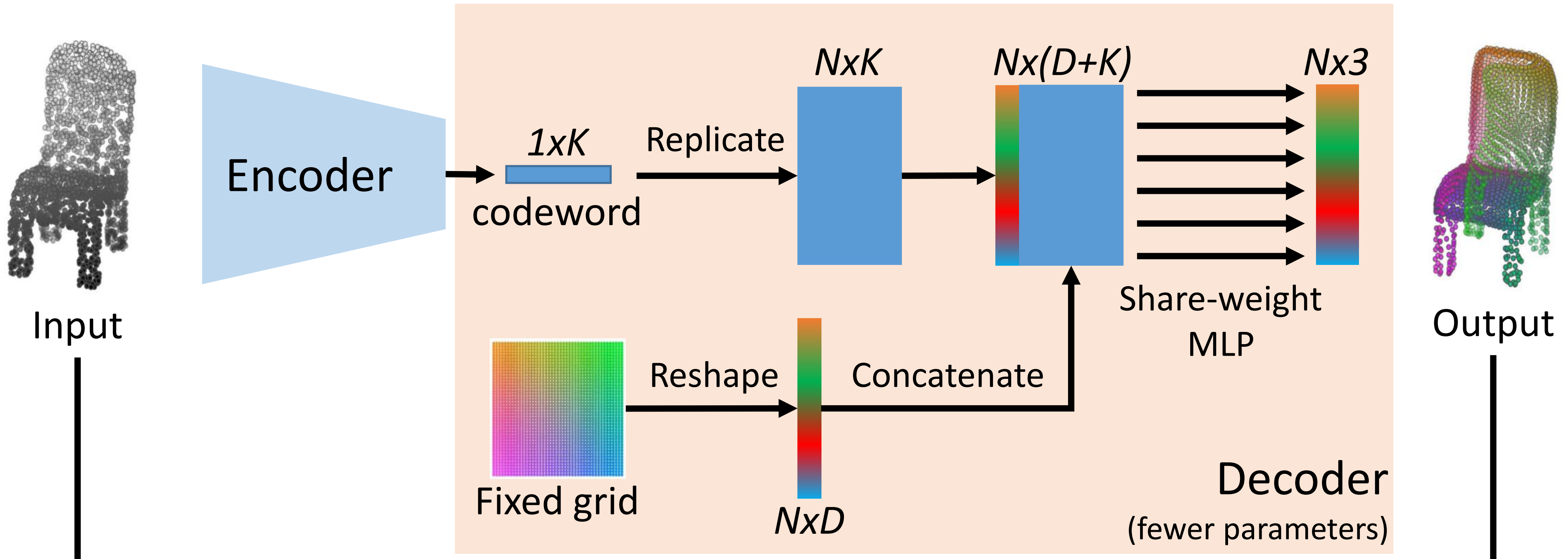

Fold


Tear


Stretch


Glue

FoldingNet Auto-encoder Framework

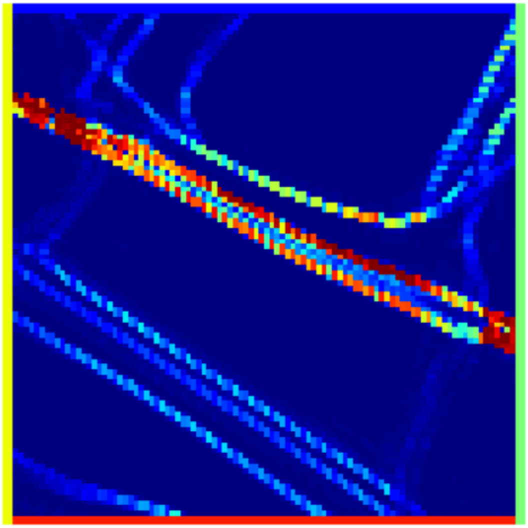


Chamfer Distance (or EMD, etc.)

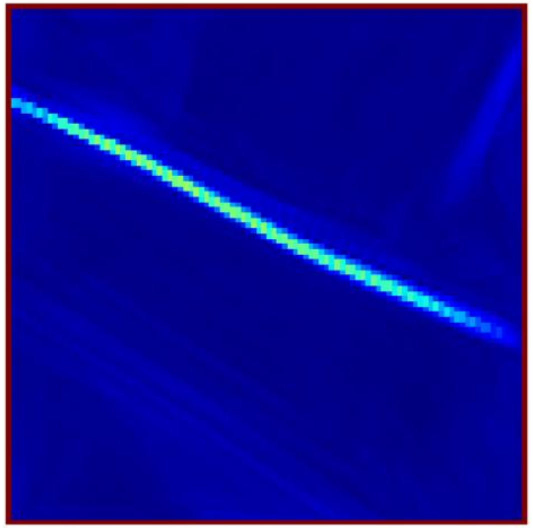
$$\max\left\{ \frac{1}{|S|} \sum_{\mathbf{x}\in S} \min_{\widehat{\mathbf{x}}\in\widehat{S}} \|\mathbf{x}-\widehat{\mathbf{x}}\|_2,\ \frac{1}{|\widehat{S}|} \sum_{\widehat{\mathbf{x}}\in\widehat{S}} \min_{\mathbf{x}\in S} \|\widehat{\mathbf{x}}-\mathbf{x}\|_2 \right\}$$

Learned Folding Profile - Sofa

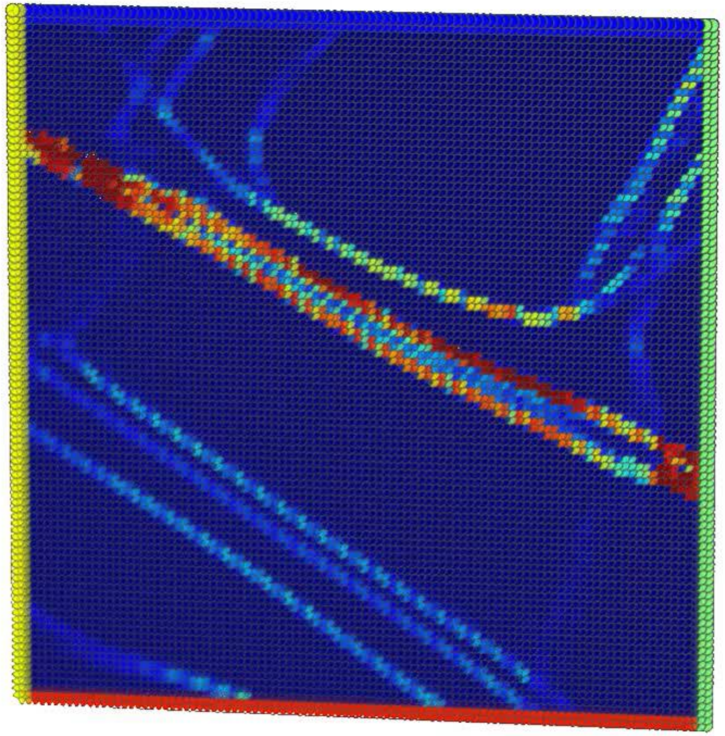
Folding Creases (Curvature)



Tear/Stretch (Neighbor Distance)



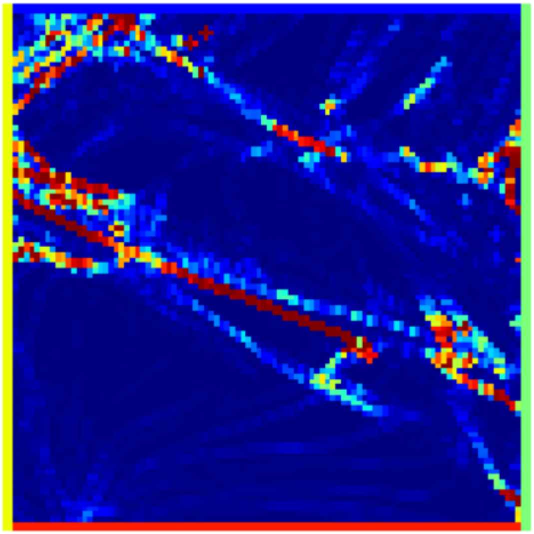
Folding Animation: Sofa
(colored by curvature)



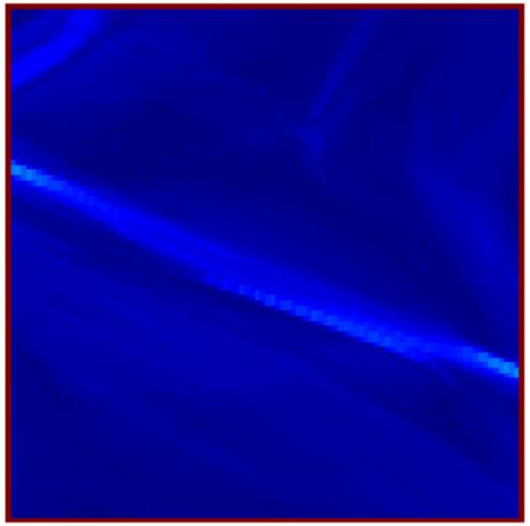
Videos of the slides available at:
https://www.youtube.com/watch?v=x1dAV4tP2oo

Learned Folding Profile - Airplane

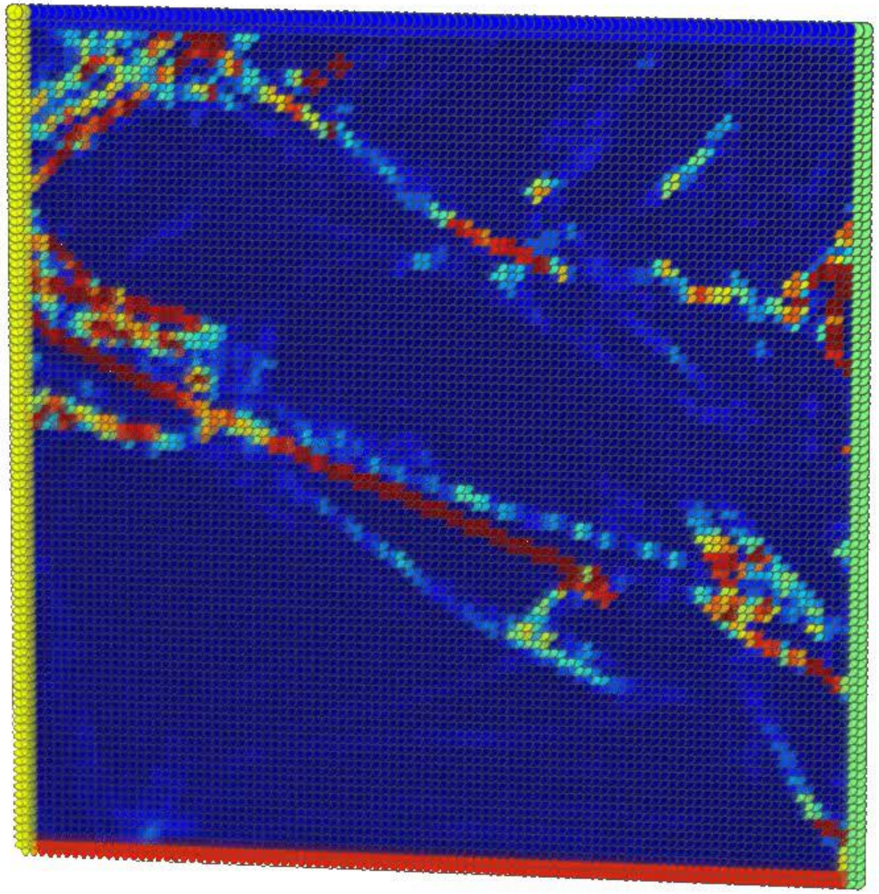
Folding Creases (Curvature)



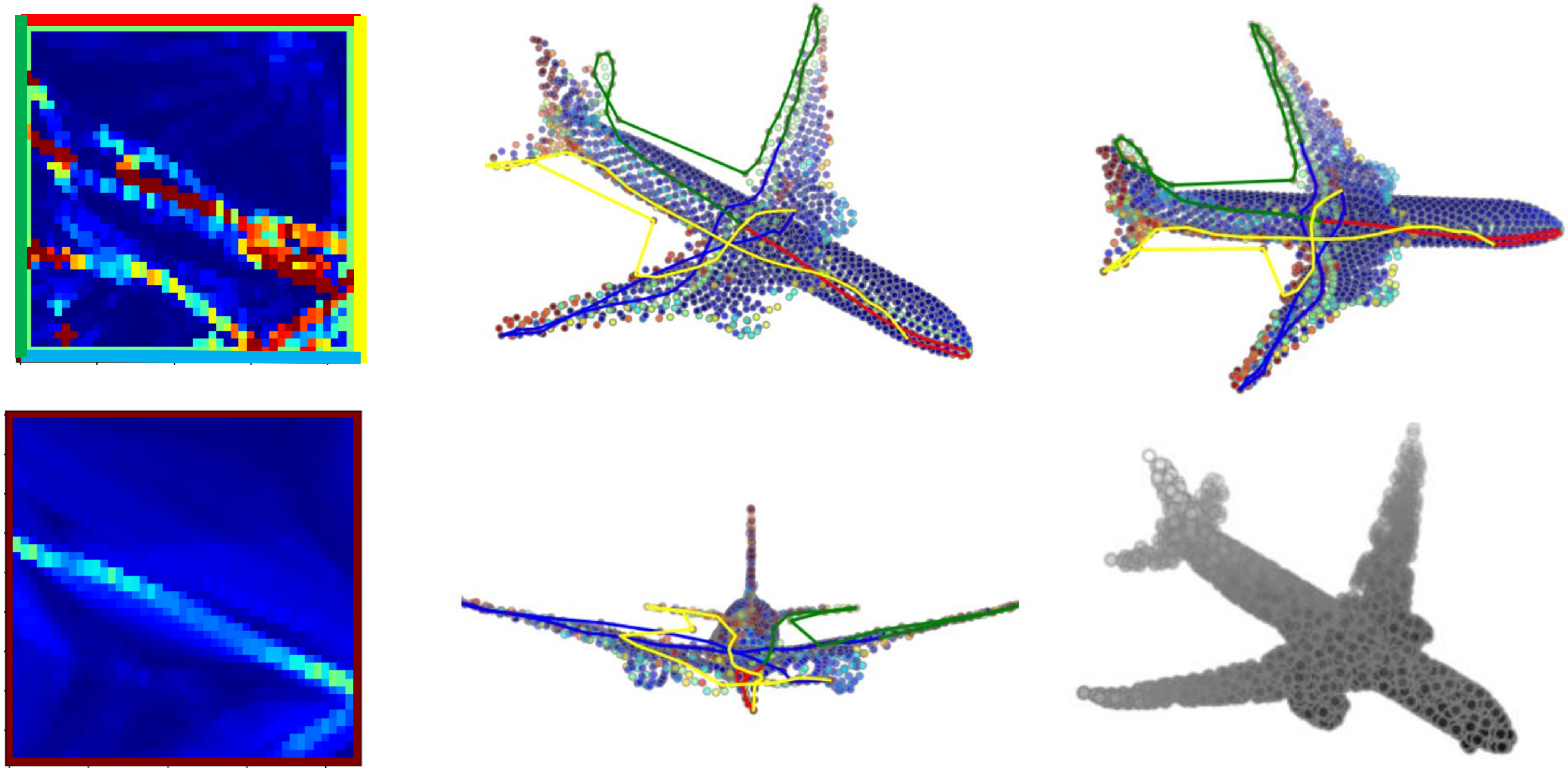
Stretch (Neighbor Distance)



Folding Animation: Airplane
(colored by curvature)

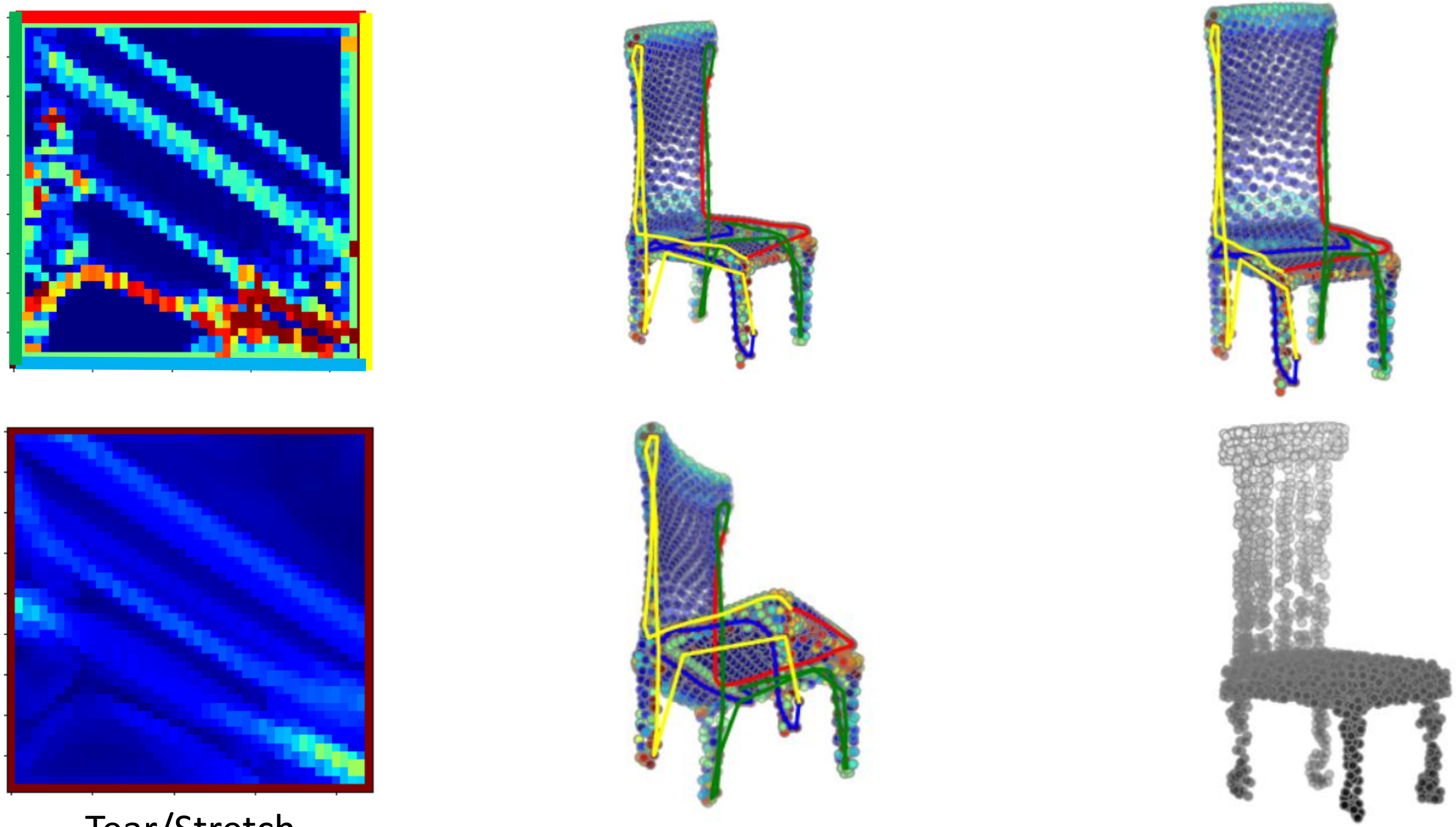


Learned Folding Profile - Airplane



Tear/Stretch

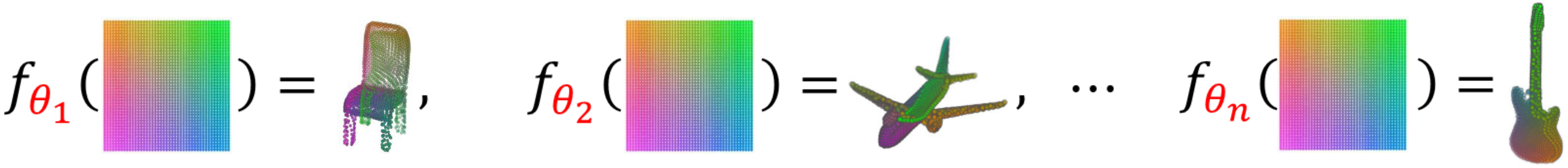
Learned Folding Profile - Chair



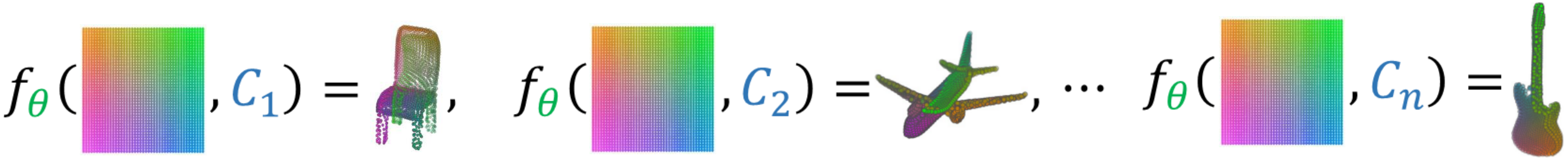
Tear/Stretch

But, can **one** DNN approx. **multiple** 2D-3D mappings?

- Universal Approximation Theorem directly tells us:
  - A specific 2-layer MLP can approximate a specific 2D-3D mapping.



- Our theorem says:
  - A single 2-layer MLP can be "tuned" by the input "codeword" to approximate multiple arbitrary 2D-3D mappings.



FoldingNet Experiments

- Training Process Visualization
- Codeword Space Visualization
- Shape Interpolation
- Transfer Classification
- Semi-supervised Learning
- Ablation Study

Training Process Visualization

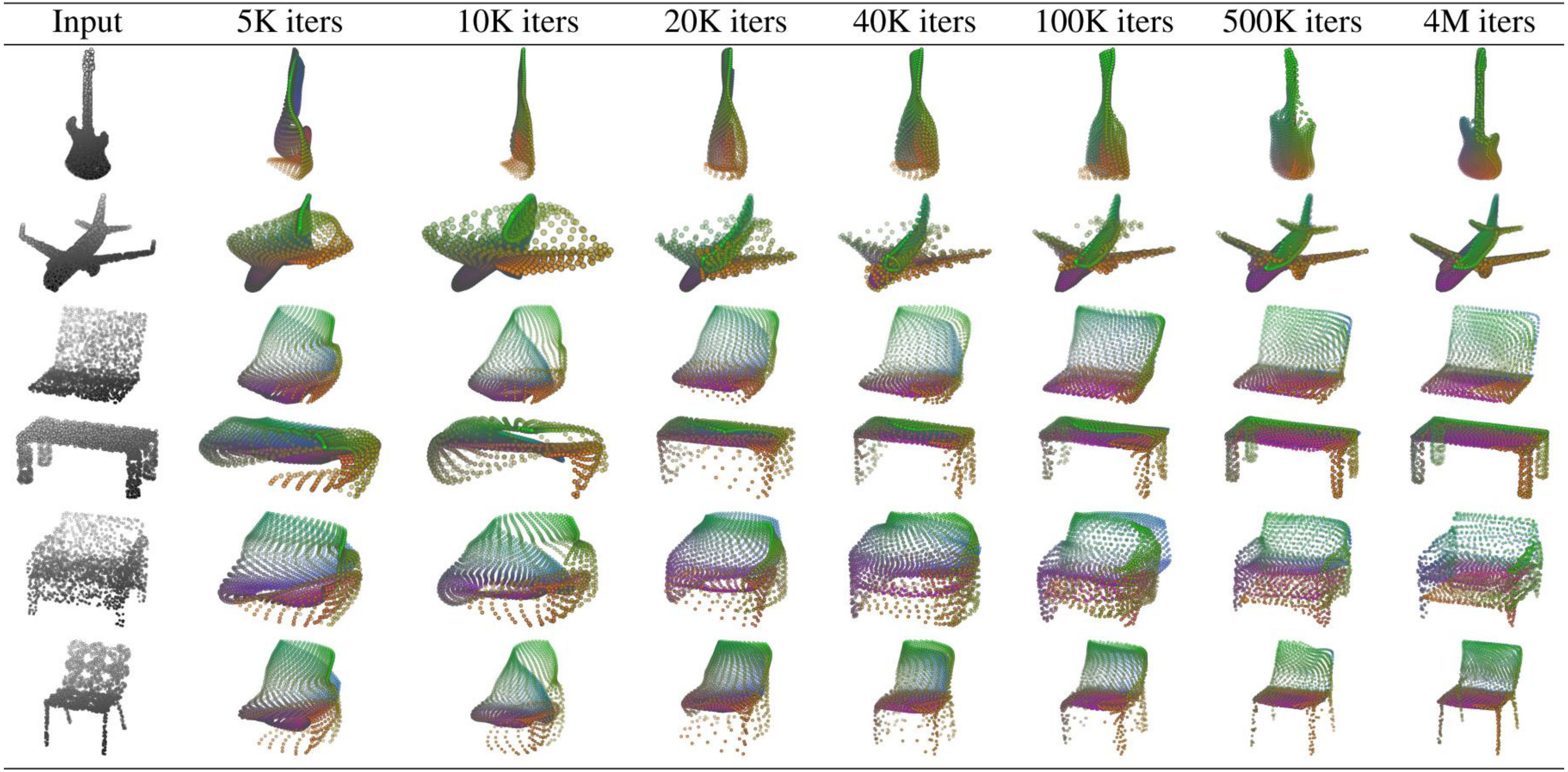


Table 2. Illustration of the training process. Random 2D manifolds gradually transform into the surfaces of point clouds.

Training Process Video

Videos of the slides available at:
https://www.youtube.com/watch?v=x1dAV4tP2oo

ModelNet

Codeword Space Visualization

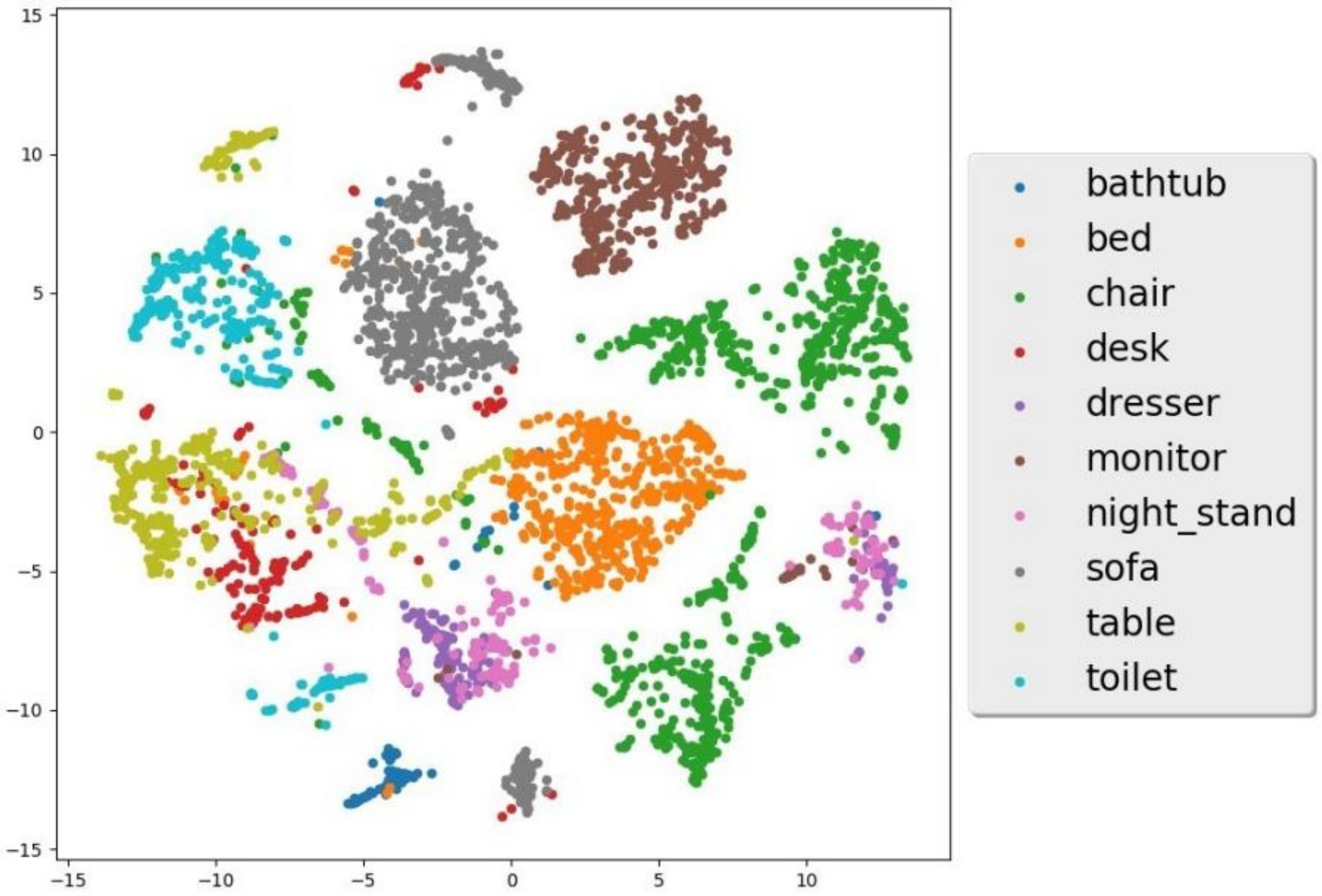
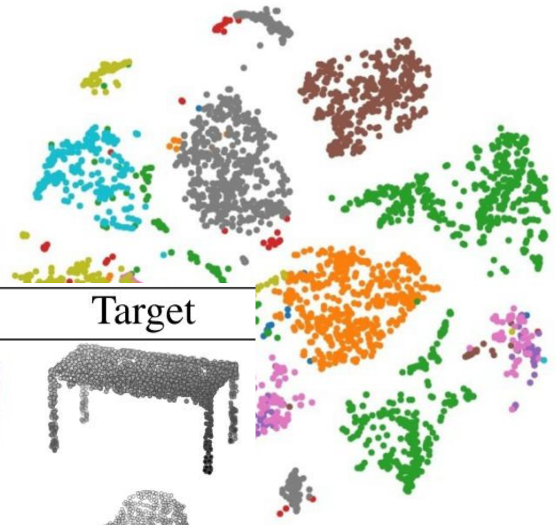


Figure 2. The T-SNE clustering visualization of the codewords obtained from FoldingNet auto-encoder.

Shape Interpolation



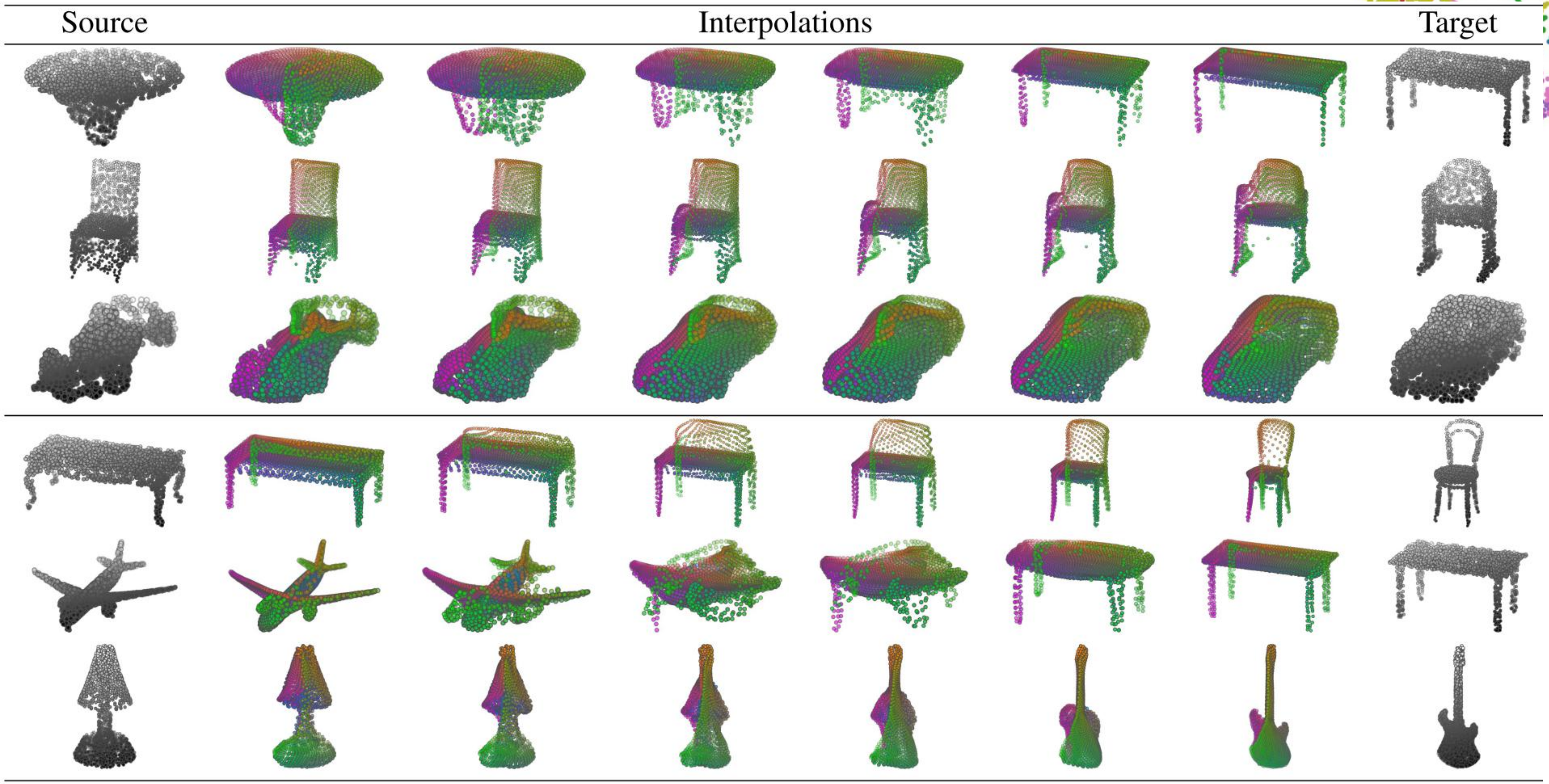
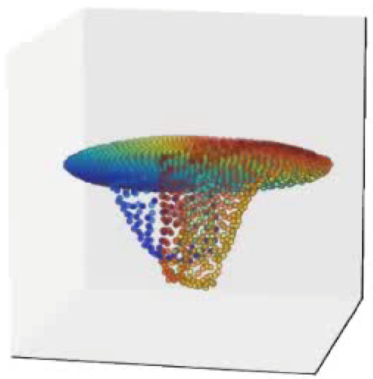


Table 3. Illustration of point cloud interpolation. The first 3 rows: intra-class interpolations. The last 3 rows: inter-class interpolations.

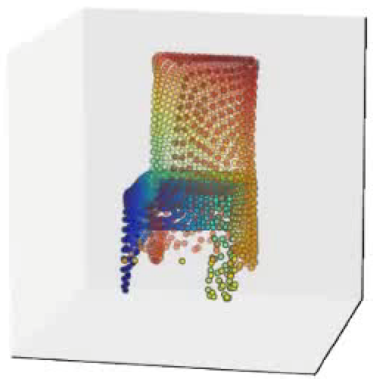
Shape Interpolation Video

Videos of the slides available at:
https://www.youtube.com/watch?v=x1dAV4tP2oo

table to table



chair to chair



car to car

car to car

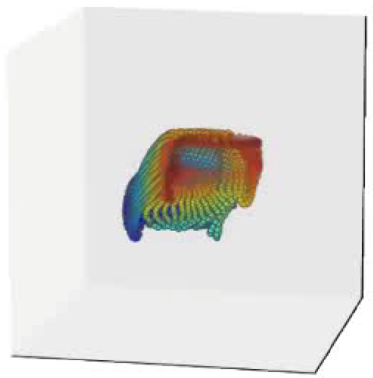


table to table

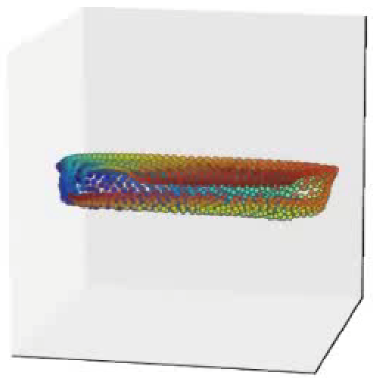
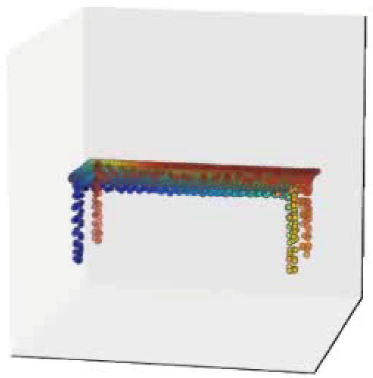
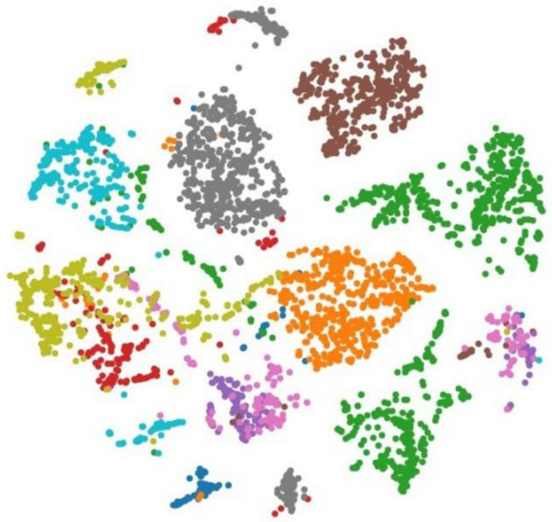


table to table

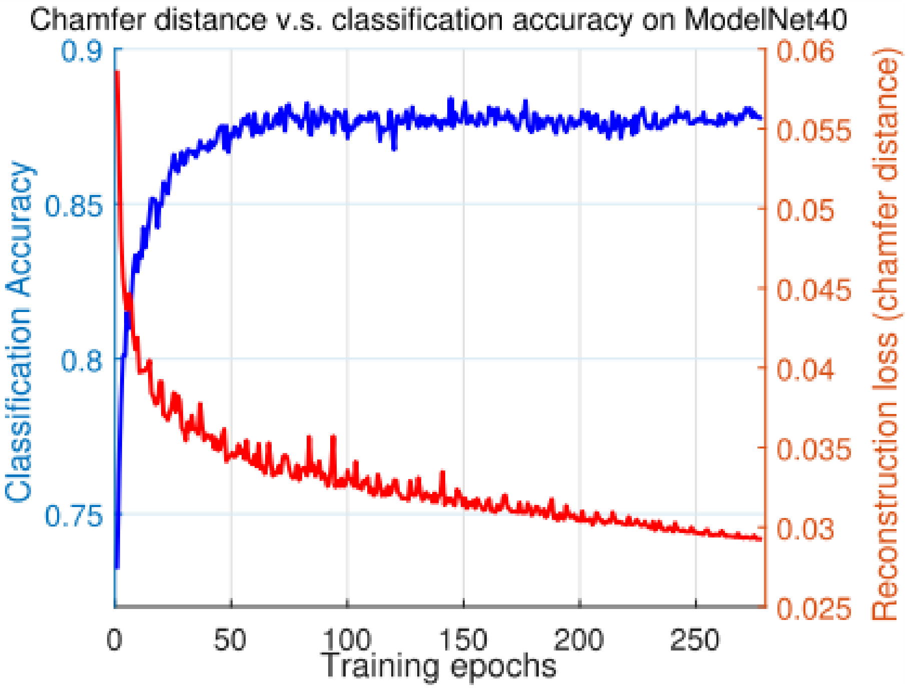


Transfer Classification

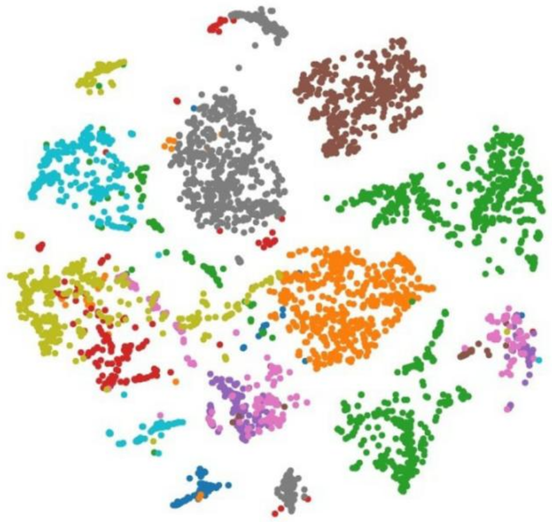


Method	MN40	MN10
SPH [26]	68.2%	79.8%
LFD [8]	75.5%	79.9%
T-L Network [19]	74.4%	-
VConv-DAE [45]	75.5%	80.5%
3D-GAN [56]	83.3%	91.0%
Latent-GAN [1]	85.7%	**95.3%**
FoldingNet (ours)	**88.4%**	94.4%

Table 5. The comparison on classification accuracy between FoldingNet and other unsupervised methods. All the methods train a linear SVM on the high-dimensional representations obtained from unsupervised training.



Semi-supervised Learning



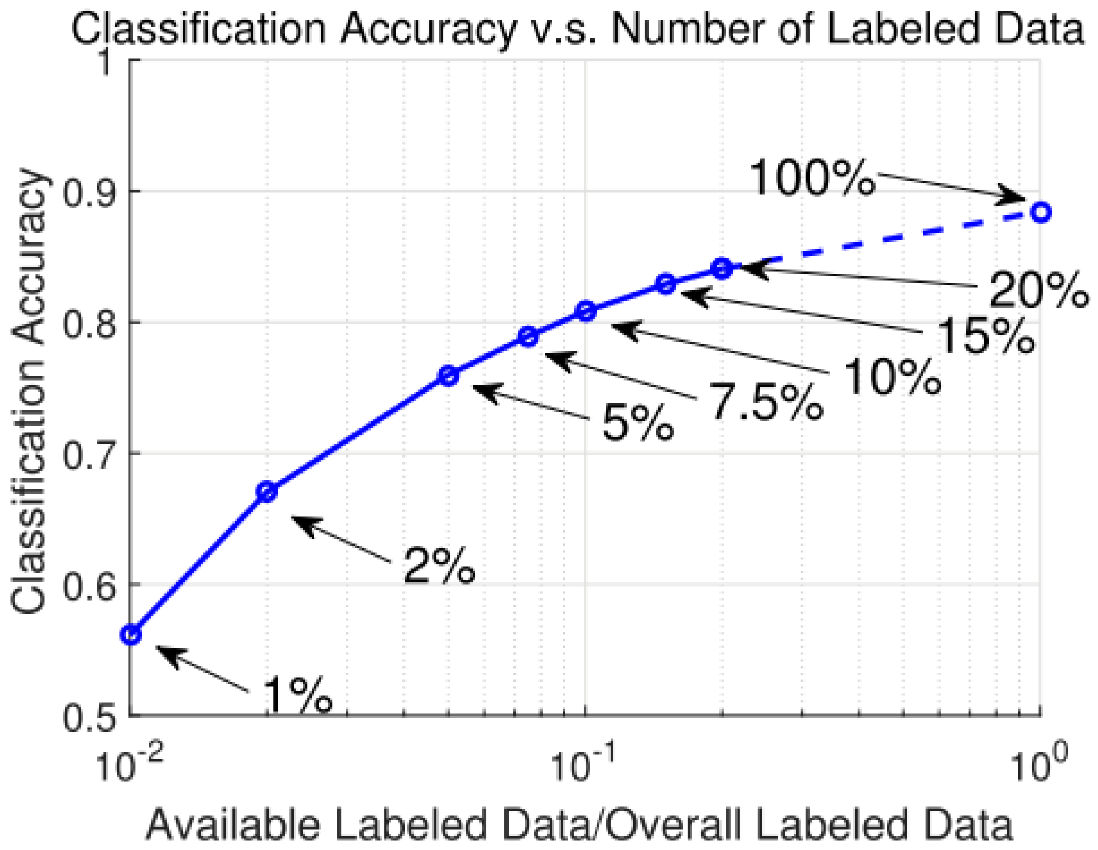


Figure 4. Linear SVM classification accuracy v.s. percentage of available labeled training data in ModelNet40 dataset.

Ablation: Decoder Variations

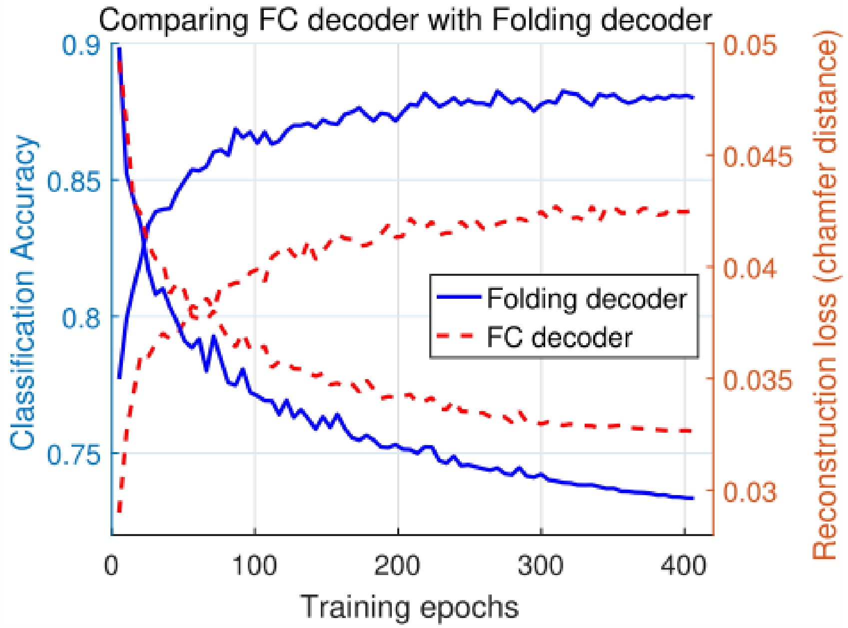


Figure 5. Comparison between the fully-connected (FC) decoder in [1] and the folding decoder on ModelNet40.

Grid Setting	#Folds	Test Cls. Acc.	Test Loss
regular 2D	2	88.25%	0.0296
regular 2D	3	88.41%	0.0290
regular 1D	2	86.71%	0.0355
regular 3D	2	88.41%	0.0284
uniform 2D	2	87.12%	0.0321

Table 6. Comparison between different FoldingNet decoders. "Uniform": the grid is uniformly random sampled. "Regular": the grid is regularly sampled with fixed spacings.

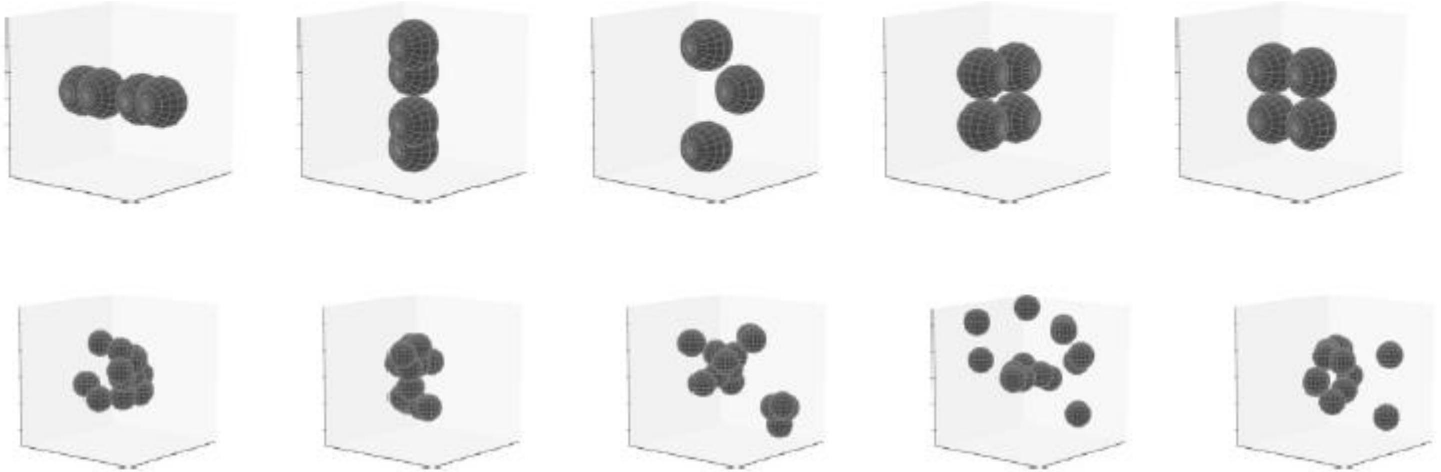
	Cl. Acc.	Tst. Loss	# Params.
FoldingNet	88.41%	0.0296	1.0×10^6
Deconv	88.86%	0.0319	1.7×10^6

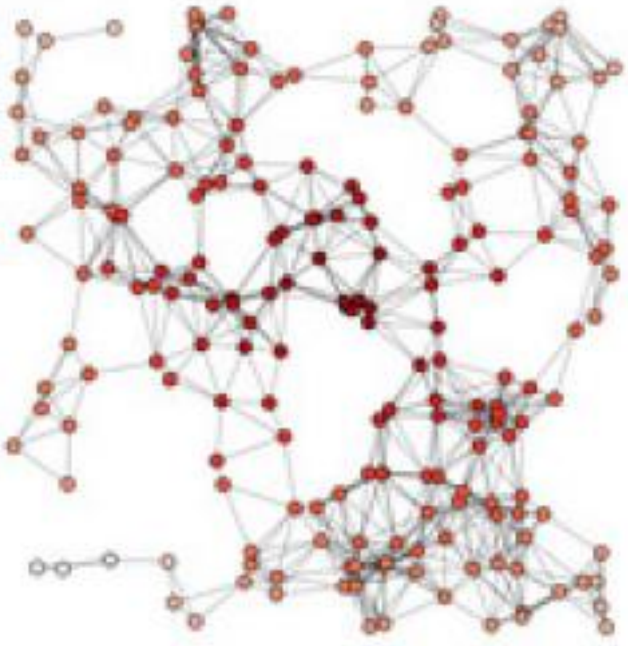
Table 7. Comparison of two different implementations of the folding operation.

Take Home Message

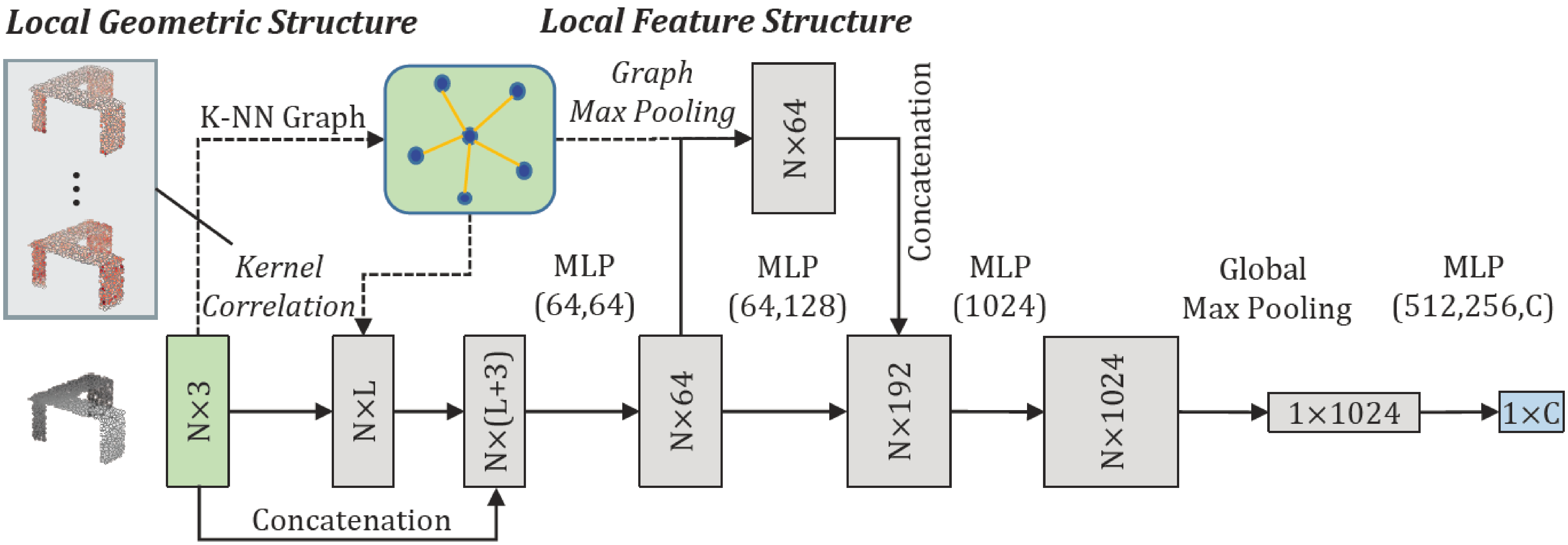
- 3D point clouds are often obtained from object surfaces
- Thus they can be transformed from one or multiple 2D planes
- FoldingNet enables data-driven learning of such transformations
- It is unsupervised: reducing labeling cost, generating point clouds
- Potential Learning-based Applications:
  - 3D Scan/Model Retrieval
  - Surface Repairing/Completion/Reconstruction
  - Scene Generation

Feature Mining on Point Clouds: Kernel Correlation and Graph Pooling



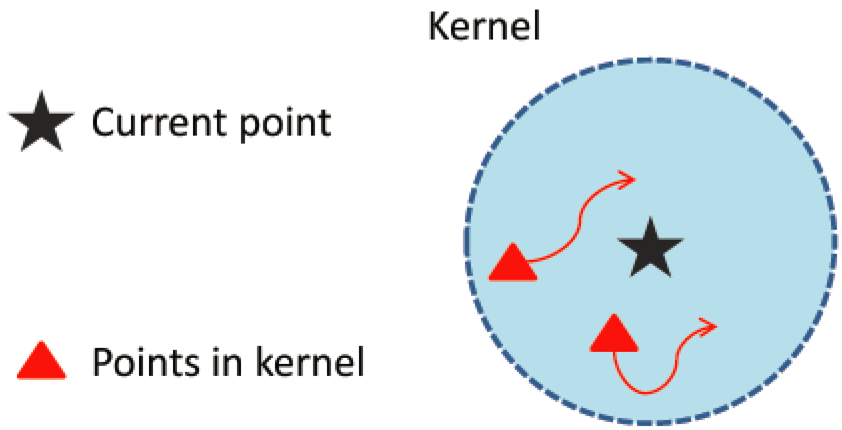


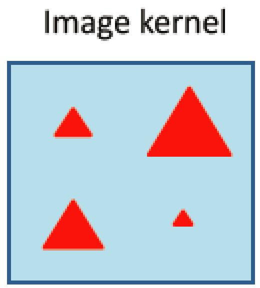
Graph-based Encoder



Learning Local Geometric Structures over Graphs

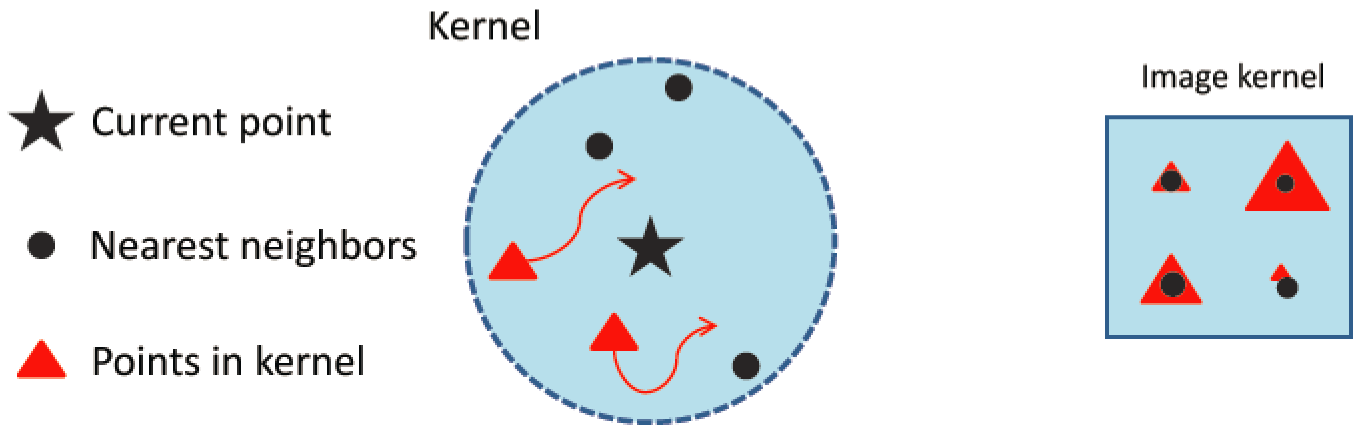
- Local geometric structures learning
  - Kernel correlation, measures geometric affinity of point sets





Learning Local Geometric Structures over Graphs

- Local geometric structures learning
  - Kernel correlation, measures geometric affinity of point sets



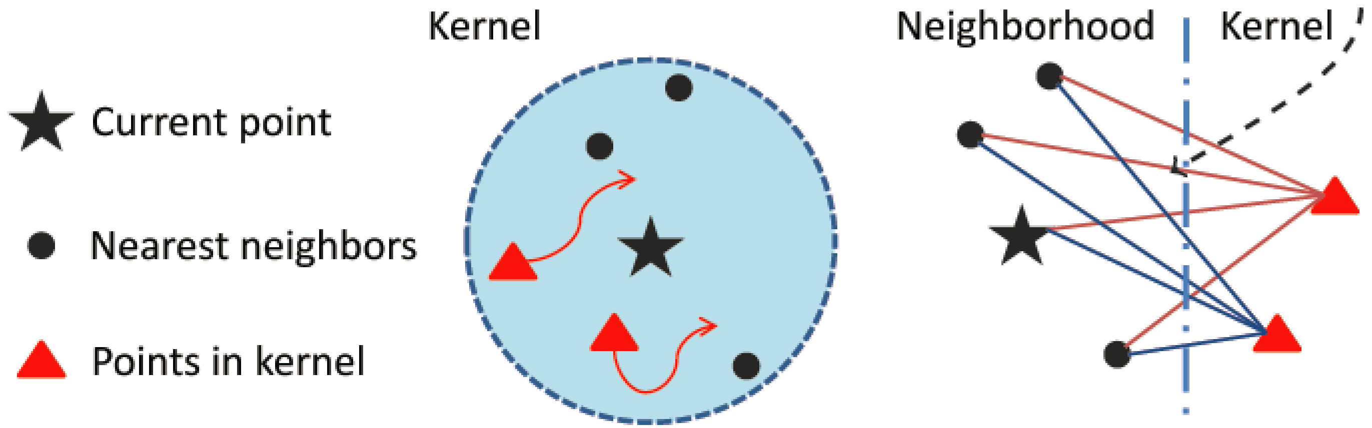
Learning Local Geometric Structures over Graphs

- Local geometric structures learning
  - Kernel correlation, measures geometric affinity of point sets

$$\mathrm{KC}(\boldsymbol{\kappa}, \mathbf{x}_c) = \frac{1}{|\mathcal{N}(\mathbf{x}_c)|} \sum_{\forall \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_c)} \sum_{k=1}^{K} \underbrace{\mathrm{K}_\sigma(\boldsymbol{\kappa}_k, \mathbf{x}_i - \mathbf{x}_c)},$$

$$\mathrm{K}_\sigma(\boldsymbol{\kappa}_k, \boldsymbol{\delta}_i) = \exp\left(-\frac{||\boldsymbol{\kappa}_k - \boldsymbol{\delta}_i||^2}{2\sigma^2}\right)$$

Graph Weights

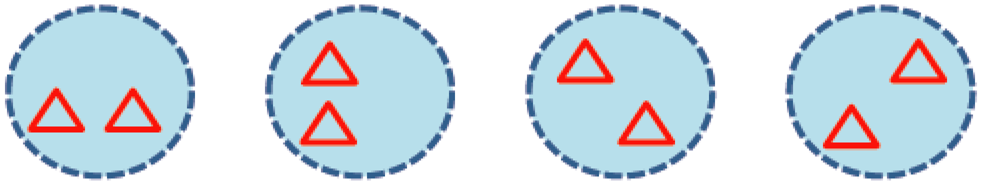


Learning Local Geometric Structures over Graphs

- Local geometric structures learning
  - Kernel correlation, measures geometric affinity of point sets

$$\mathrm{KC}(\boldsymbol{\kappa}, \mathbf{x}_c) = \frac{1}{|\mathcal{N}(\mathbf{x}_c)|} \sum_{\forall \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_c)} \sum_{k=1}^{K} \mathrm{K}_\sigma(\boldsymbol{\kappa}_k, \mathbf{x}_i - \mathbf{x}_c),$$

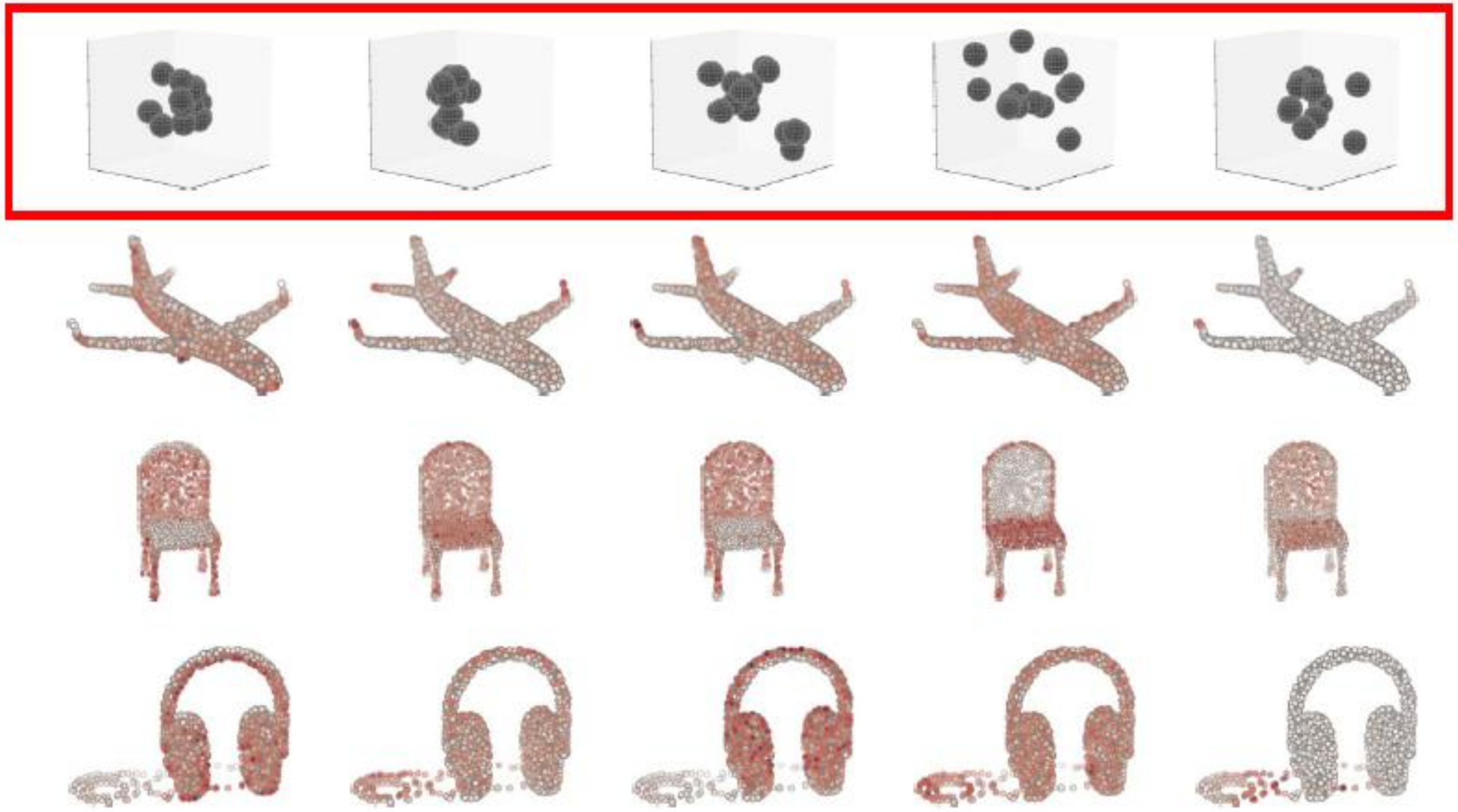
$$\mathrm{K}_\sigma(\boldsymbol{\kappa}_k, \boldsymbol{\delta}_i) = \exp\left(-\frac{||\boldsymbol{\kappa}_k - \boldsymbol{\delta}_i||^2}{2\sigma^2}\right)$$



Potential kernels learned

Learning Local Geometric Structures over Graphs

- Example kernels learned and filter responses



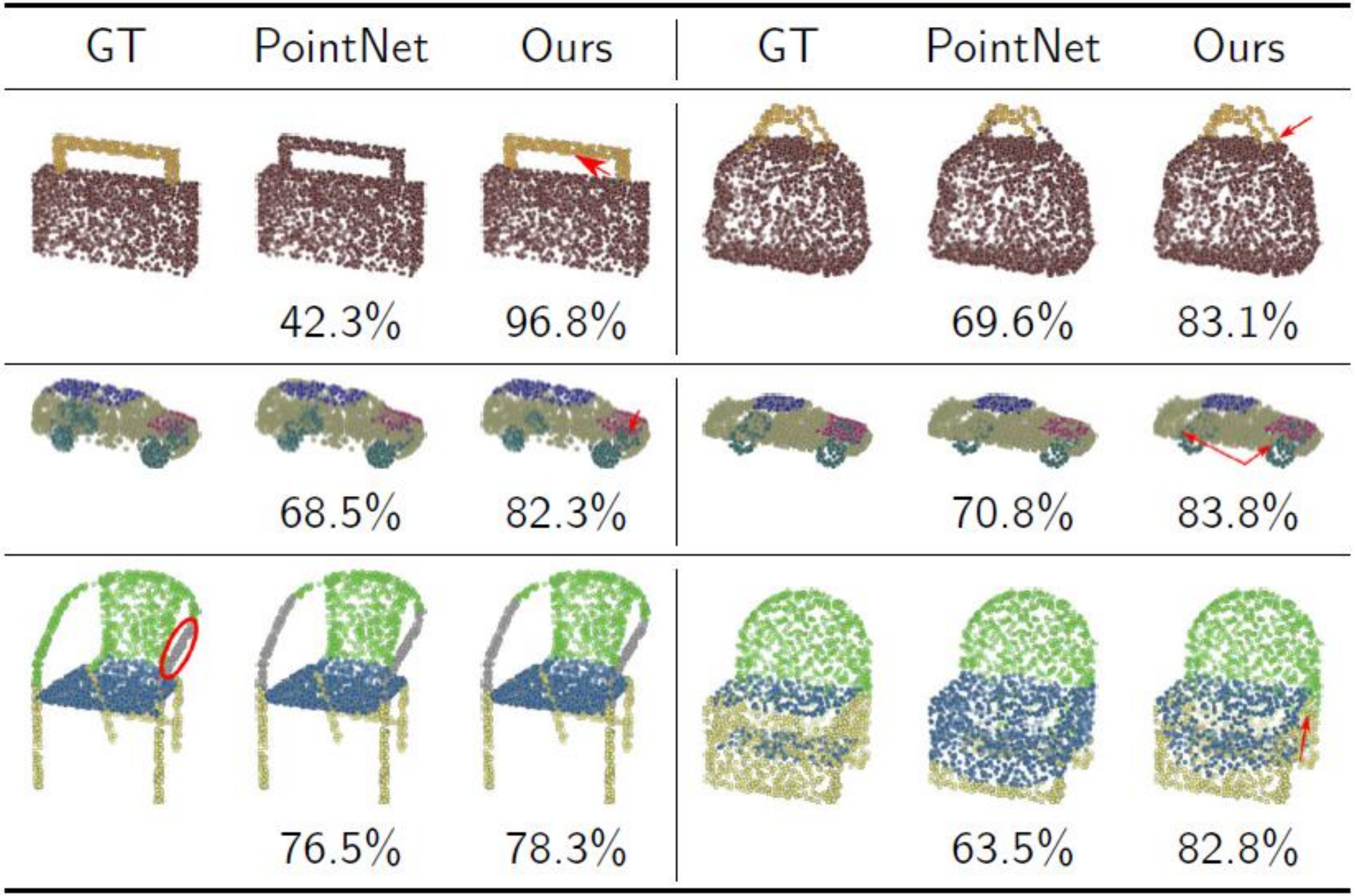
Shape Classification

- ModelNet10, ModelNet40
  - Uniformly sampling from meshes
- $L = 32$ kernels, each kernel has $K = 16$ points
- Main competing method, PointNet++
  - Slightly better accuracy with less number of parameters

Method	MN10	MN40
MVCNN [36]	-	90.1
VRN Ensemble [2]	**97.1**	**95.5**
ECC [34]	90.0	83.2
PointNet (vanilla) [29]	-	87.2
PointNet [29]	-	89.2
PointNet++ [31]	-	90.7
KCNet (ours)	**94.4**	**91.0**

Image avail as inputs

Object Part Segmentation



Take Home Message

- Find graph embedding via learning
- Graph topology 1: A global neighborhood graph
    - Graph pooling
    - Local geometry learning
    - Local feature aggregation
- Graph topology 2: Local bipartite graphs
    - Local geometry learning
    - Local feature maps

Thanks for your attention!

Any questions?