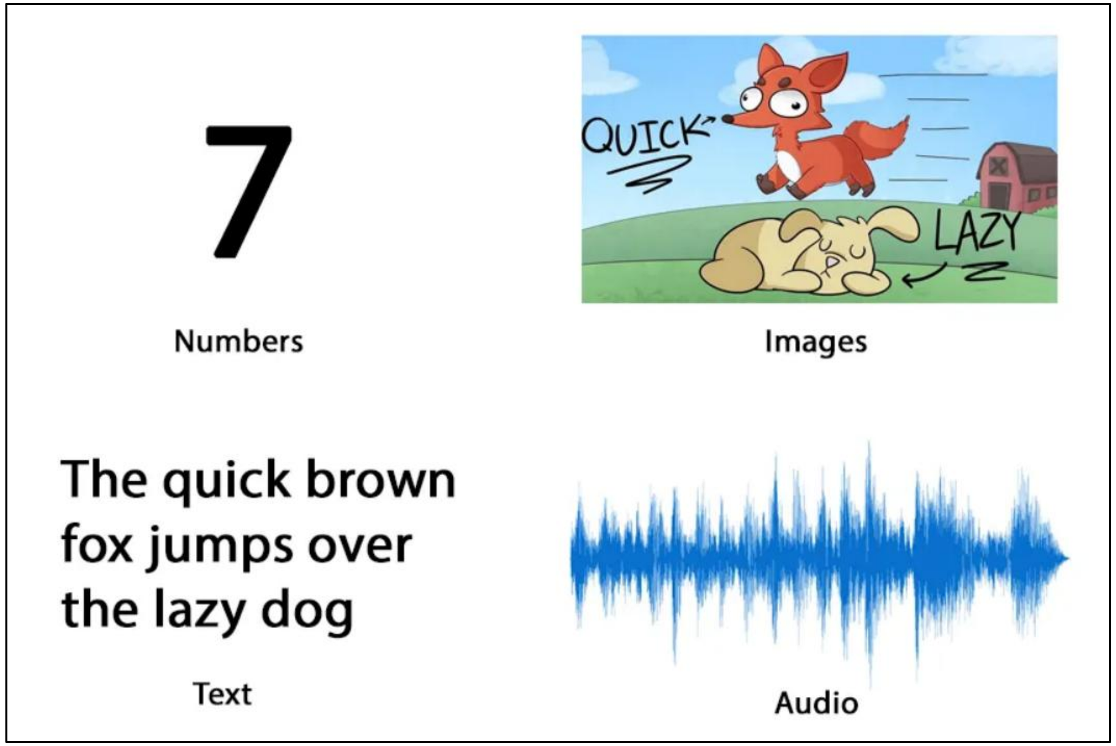
Geometric Deep Learning

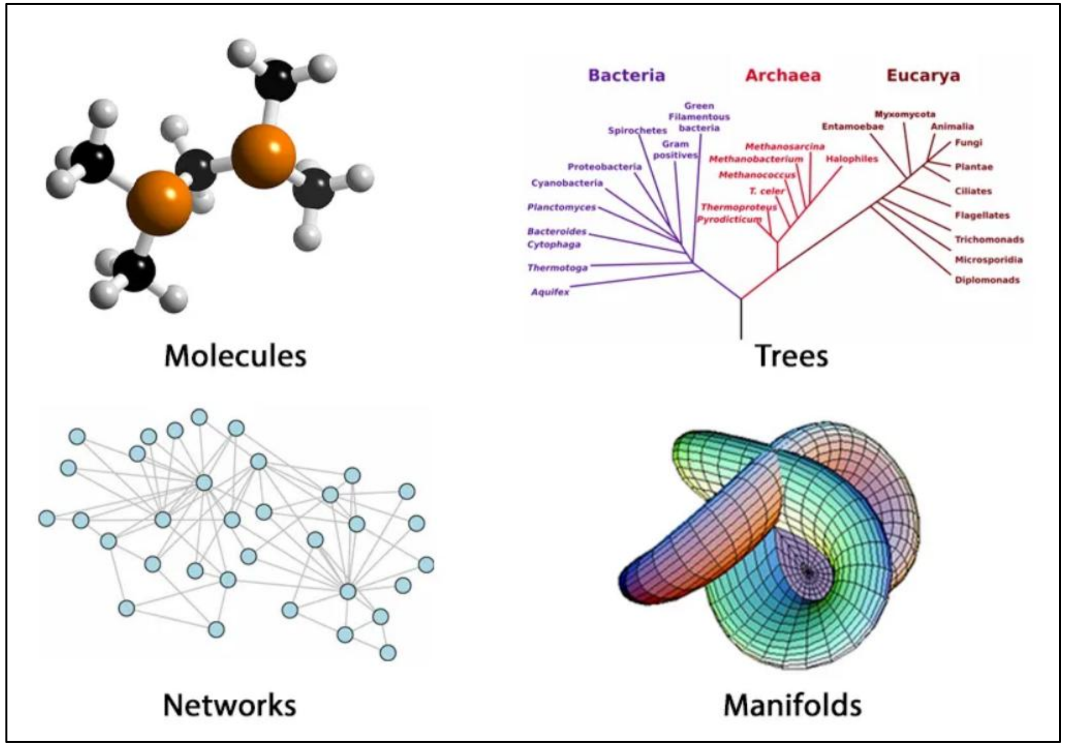
— From the Perspective of 3D Geometry Processing

几何深度学习

Geometric Deep Learning aims to generalize neural network models to non-Euclidean domains such as graphs and manifolds.



Euclidean Data

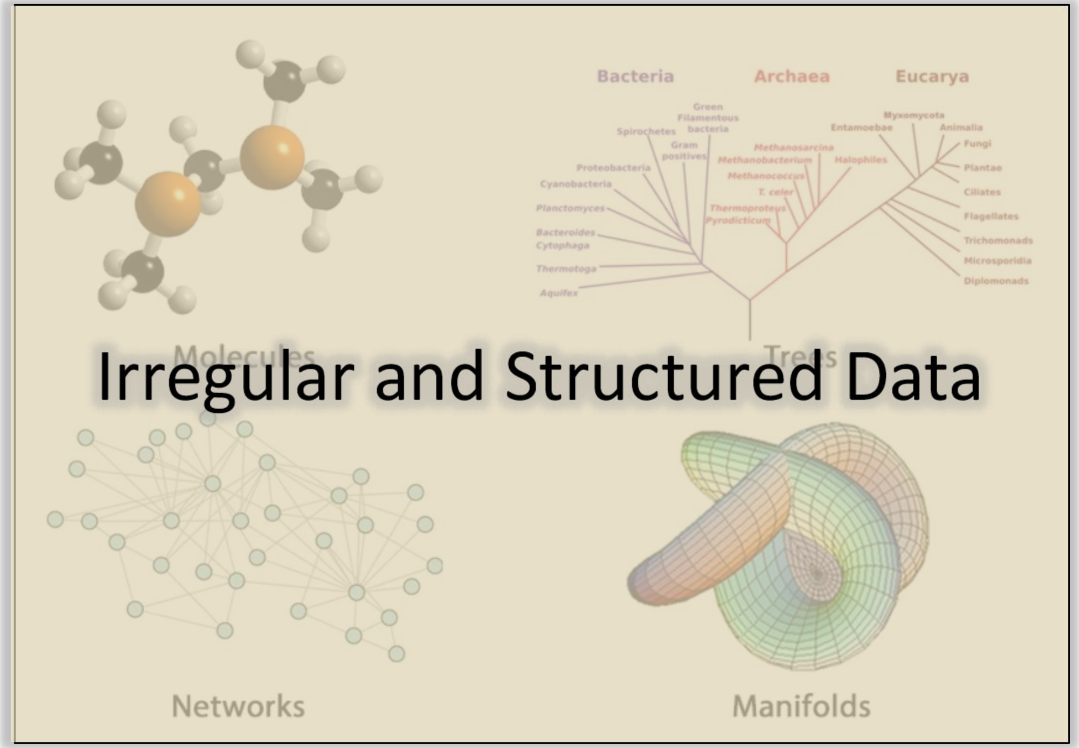


Non-Euclidean Data

几何深度学习

Geometric Deep Learning aims to generalize neural network models to non-Euclidean domains such as graphs and manifolds.

Euclidean Data

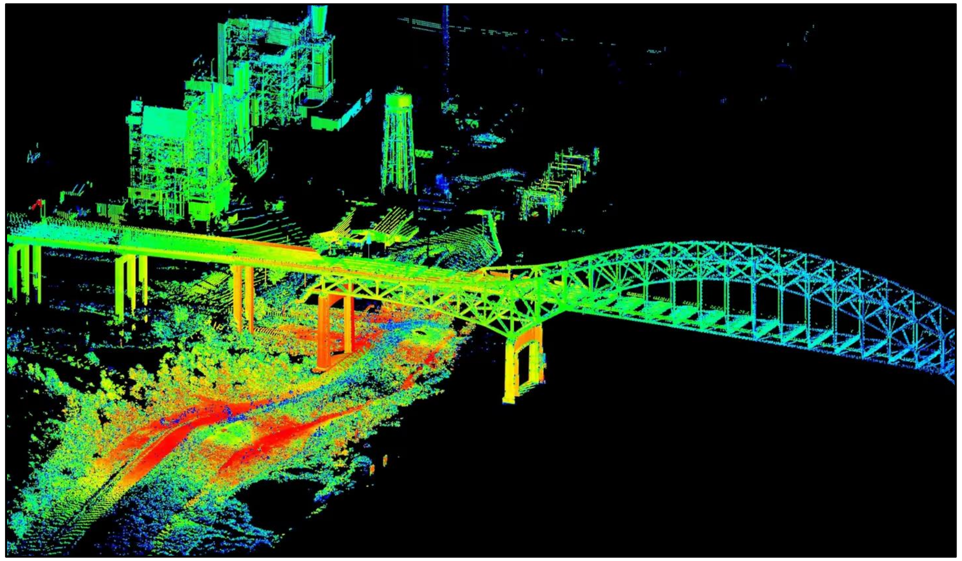


Non-Euclidean Data

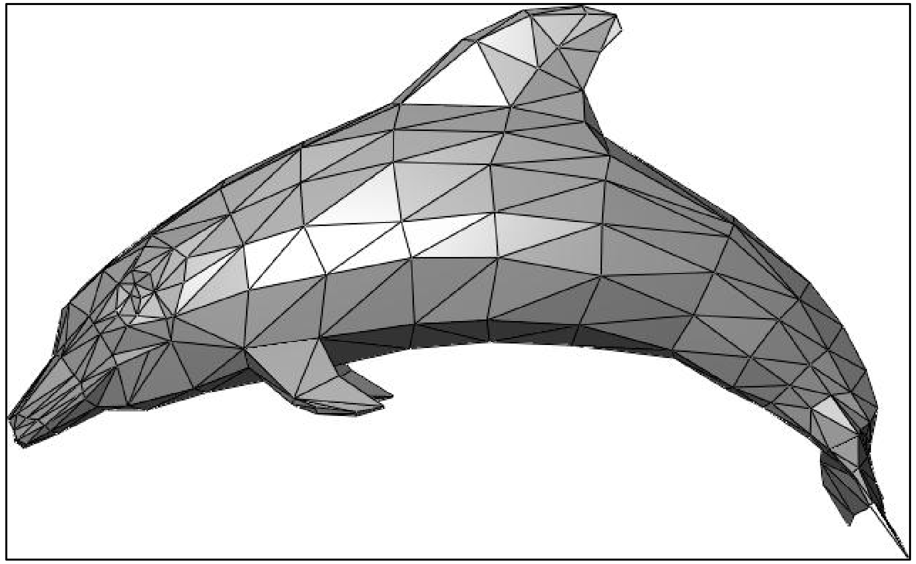
非欧几里得数据

Non-euclidean data can represent more complex items and concepts with more accuracy than 1D or 2D representation:

- Point Cloud: represented as a Nx3 array, but it's not a 2D grid!
- Mesh: represented as a list of vertices and faces



Point Cloud



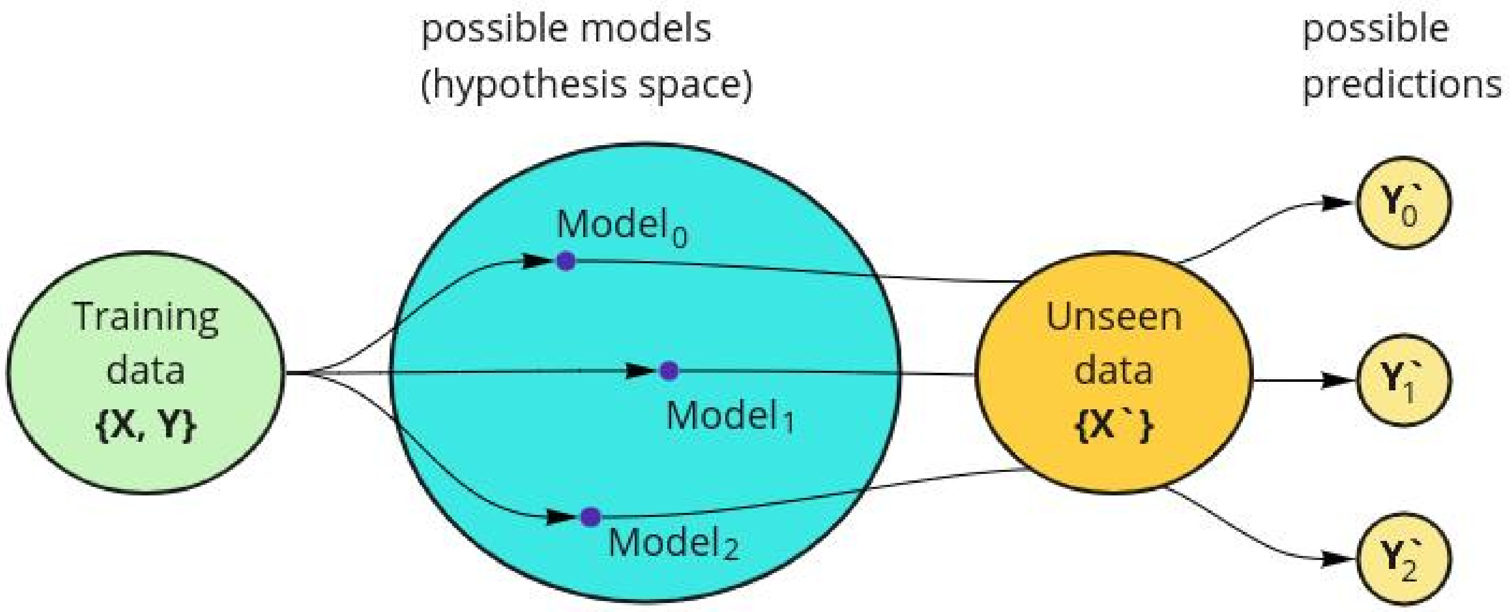
Mesh

非欧几里得数据

Inductive bias of non-euclidean data is that, given data of an arbitrary type, format, and size, one can prioritize the model to learn certain patterns by changing the structure of that data.

Inductive bias is the set of assumptions that a machine learning algorithm makes about the relationship between input variables (features) and output variables (labels) based on the training data.

—— Mitchell, 1980



Inductive Bias

非欧几里得数据

Fundamentally, geometric deep learning invovles encoding a geometric understanding of data as an inductive bias in deep learning models to give them a helping hand.

Three types of geometric priors:

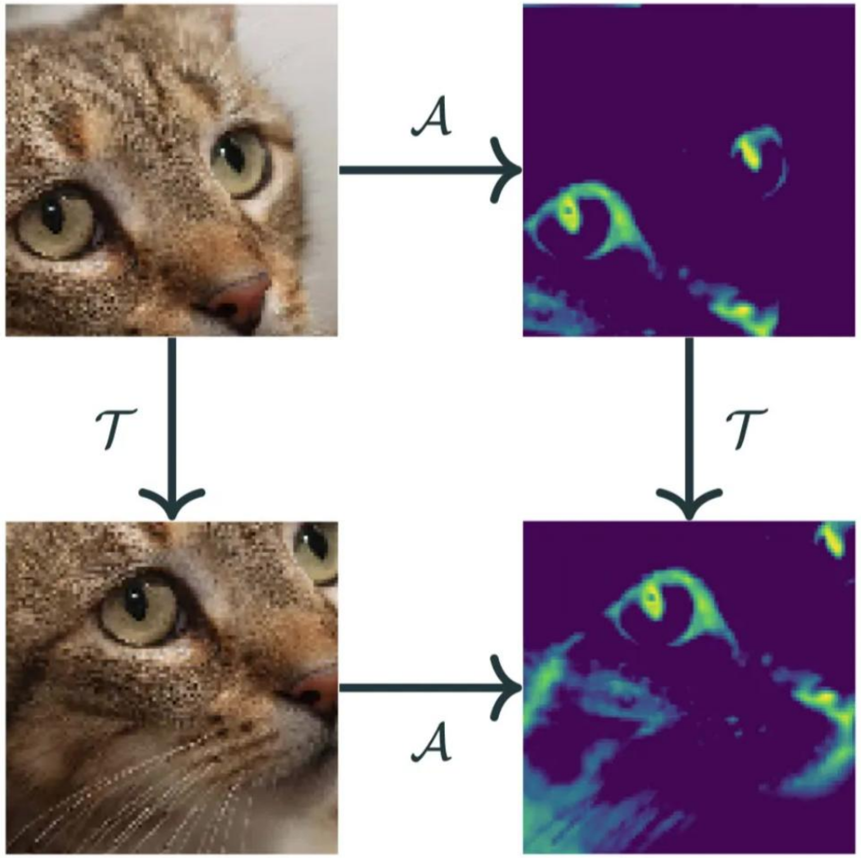
- Symmetry and invariance
- Stability
- Multiscale representations

非欧几里得数据

Fundamentally, geometric deep learning invovles encoding a geometric understanding of data as an inductive bias in deep learning models to give them a helping hand.

Three types of geometric priors:

- **Symmetry and invariance**
- Stability
- Multiscale representations



Translational Equivariance

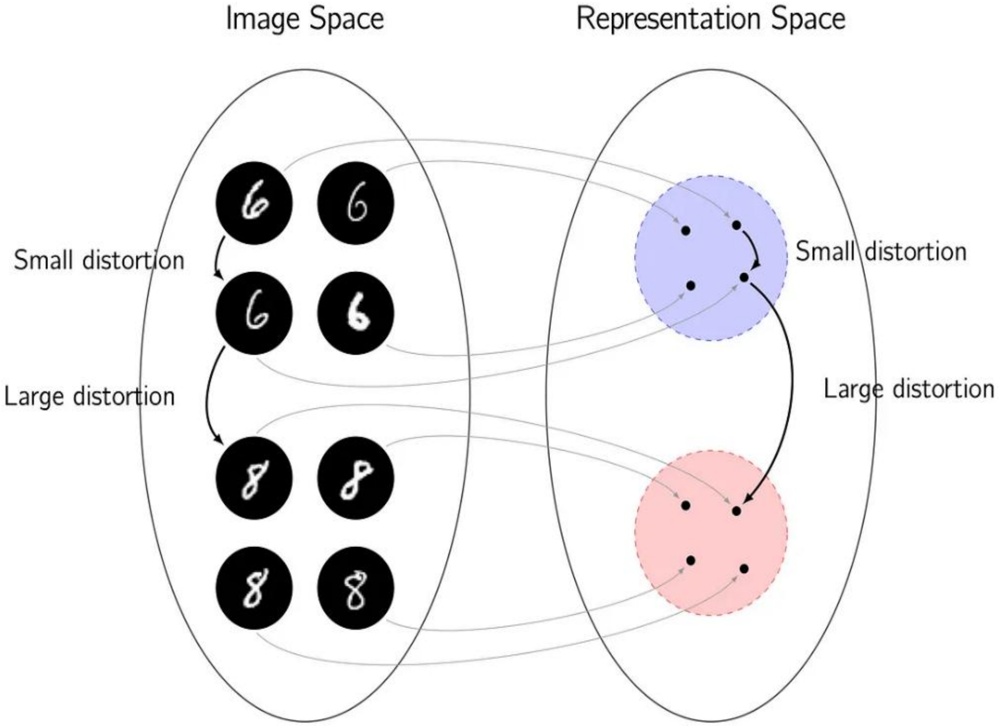
Computing a feature map (by $\mathcal{A}$) (top right) and then translating ($\mathcal{T}$) the feature map (bottom right) is equivalent to first translating the image (bottom left) and then computing the feature map

非欧几里得数据

Fundamentally, geometric deep learning invovles encoding a geometric understanding of data as an inductive bias in deep learning models to give them a helping hand.

Three types of geometric priors:

- Symmetry and invariance
- **Stability**
- Multiscale representations



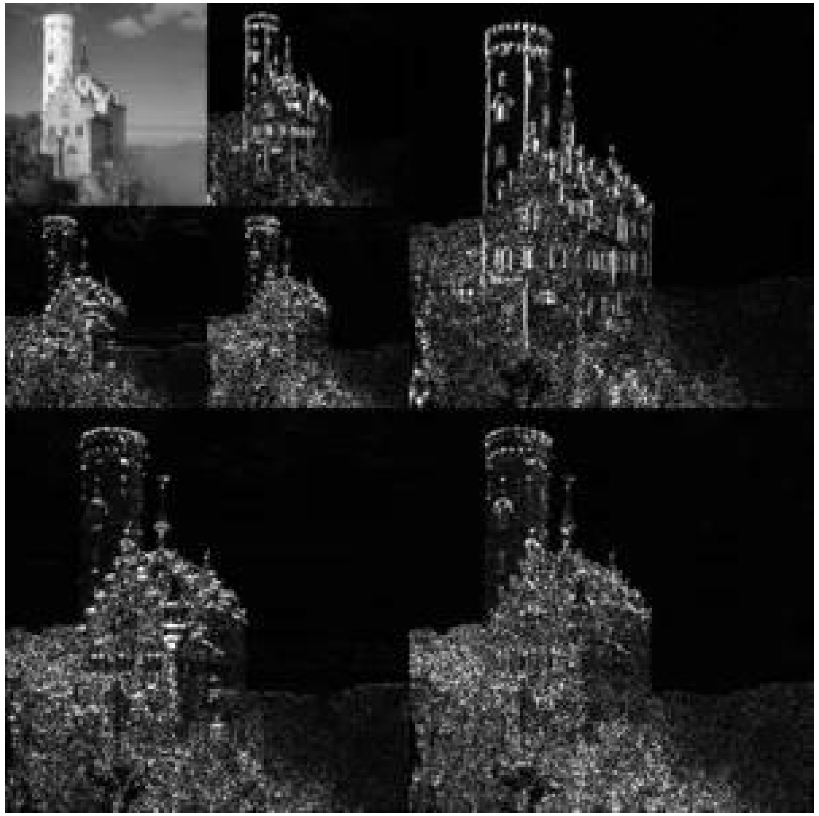
Small distortions are responsible for intra-class variations, whereas large distortions are responsible for inter-class variations. Stability of the mapping is required to **ensure measures of similarity between data instances.**

非欧几里得数据

Fundamentally, geometric deep learning invovles encoding a geometric understanding of data as an inductive bias in deep learning models to give them a helping hand.

Three types of geometric priors:

- Symmetry and invariance
- Stability
- **Multiscale representations**

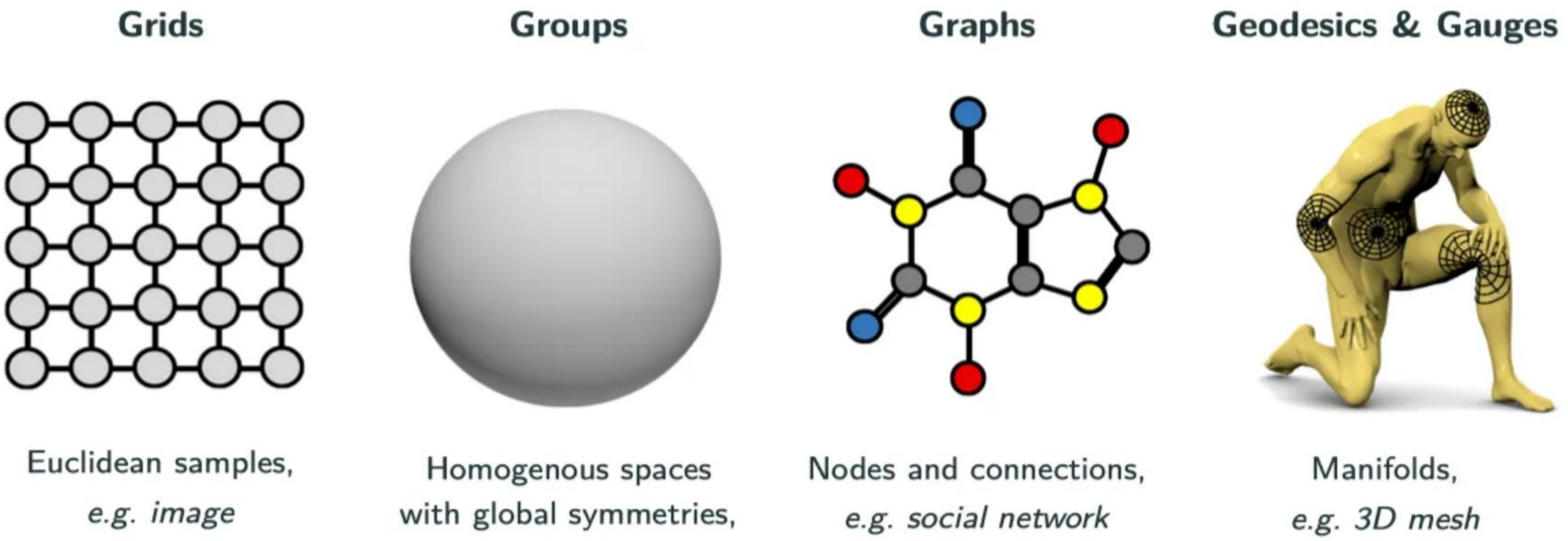


Multiscale and Hierarchical representation provides the global content in the low-resolution version and detailed local information in the high-resolution version.

方法分类

Geometric deep learning is classified into four fundamental categories

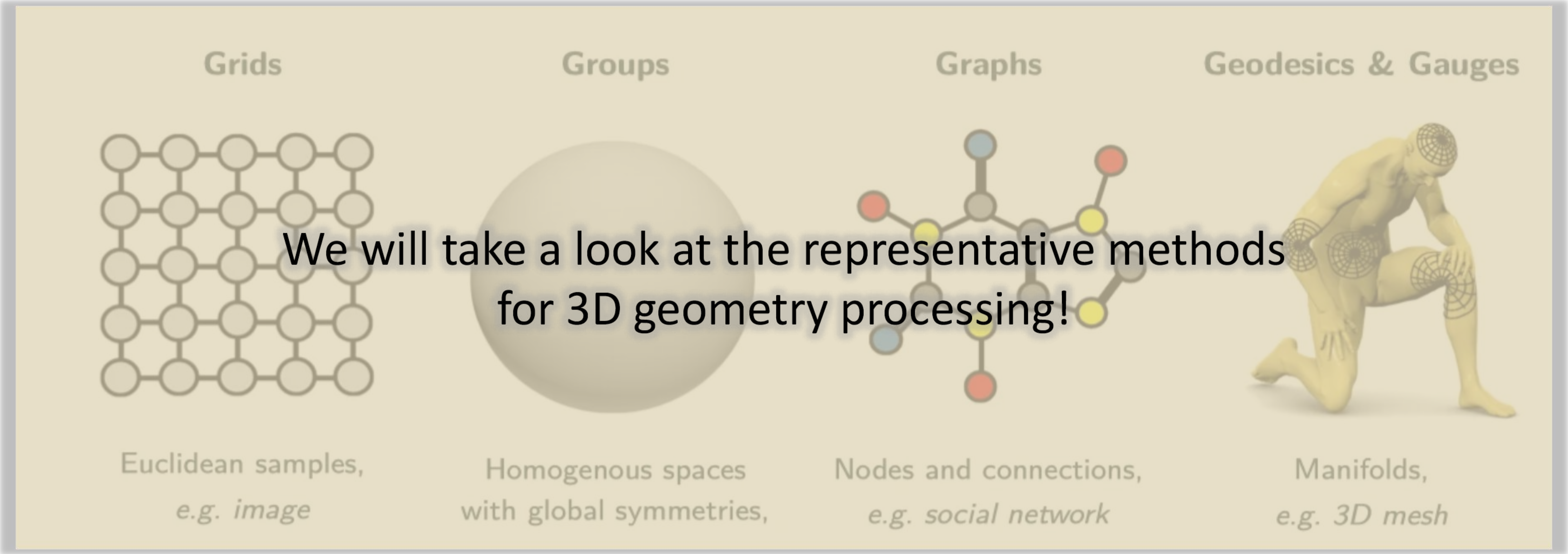
— Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges (2021)



方法分类

Geometric deep learning is classified into four fundamental categories

— Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges (2021)

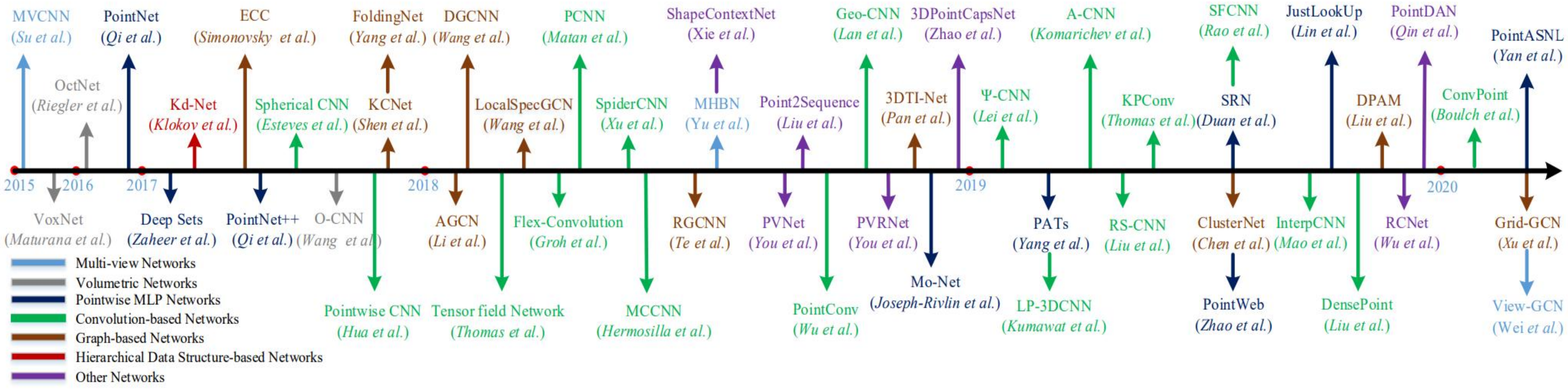


We will take a look at the representative methods for 3D geometry processing!

Geometric Deep Learning for 3D Point Clouds

三维点云学习方法总览

- There has been a vast of deep learning methods for 3D point cloud processing, e.g. shape classification, object detection and tracking, 3D segmentation.



A chronological overview of deep learning-based 3D shape classification methods

Deep Learning for 3D Point Clouds: A Survey

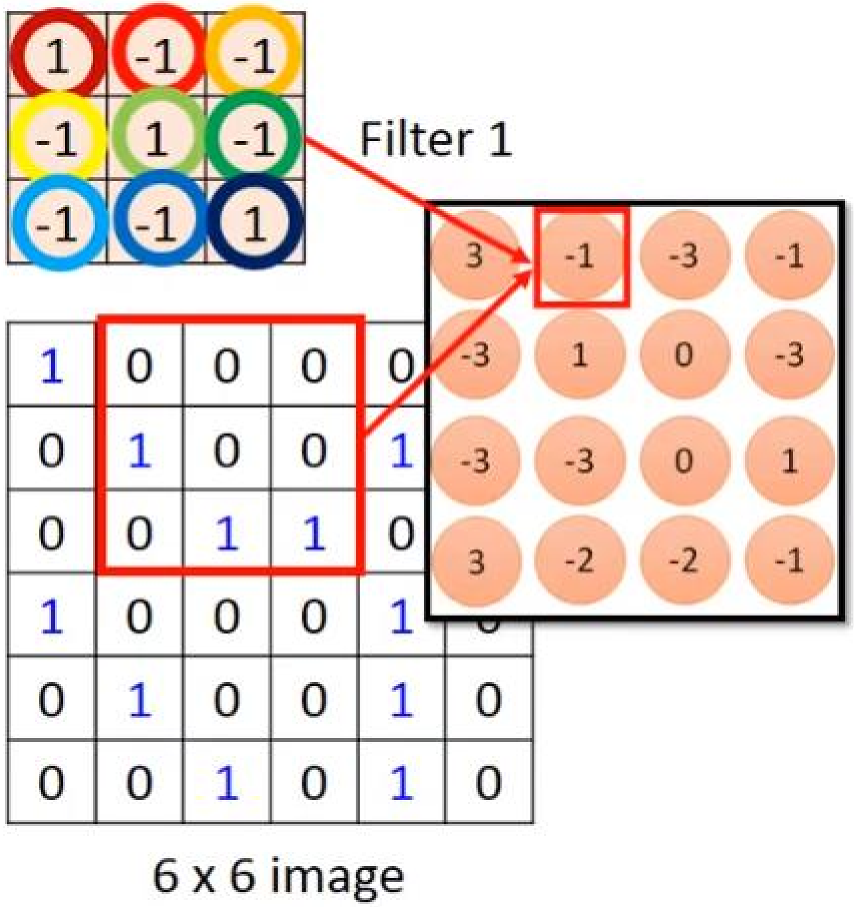
三维点云的几何特性

Point Set Surface

- **Unordered.** Point cloud is a set of points without specific order.
- **Interaction among points.** Points are not isolated, and neighboring points form a meaningful subset.
- **Invariance under transformations.** The learned representation of the point set should be invariant to certain transformations.

回顾卷积神经网络

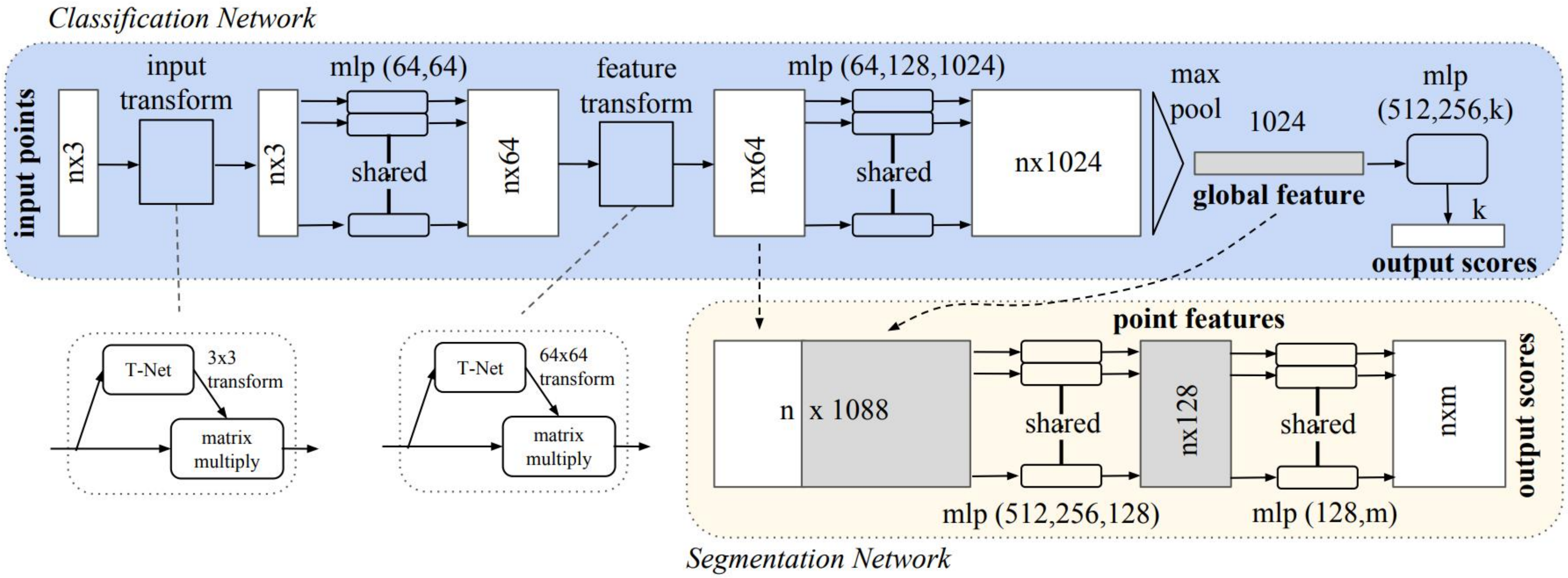
2D Convolutional Neural Network



Aggregation Function
The convolution operator aggregates the features within a sliding window into a higher-level feature.

PointNet, 2017

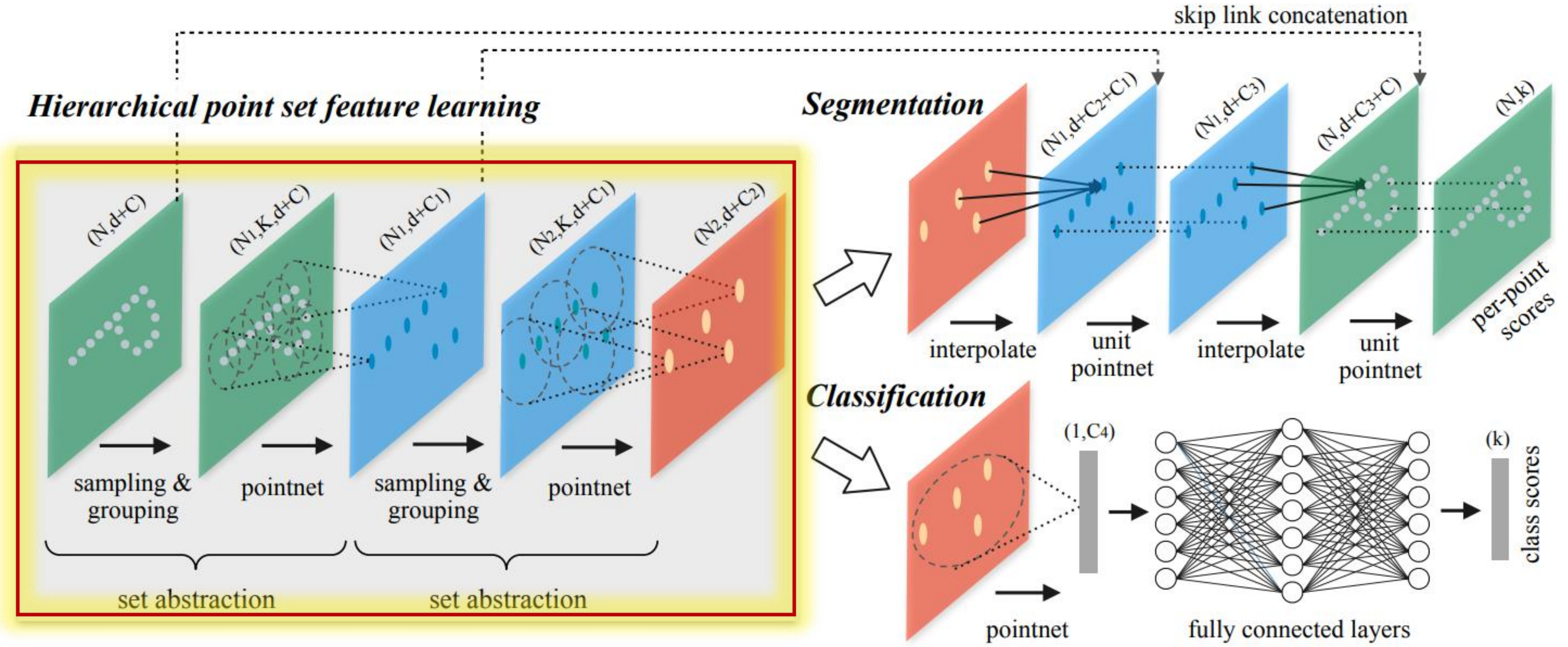
- Fully-connected layers to encode the shape feature.
- A maxpooling aggregation function for the unordered input.
- Canonical alignment to be invariant to transformations.



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

PointNet++, 2017

- Progressively encode the shape feature in a coarse-to-fine manner

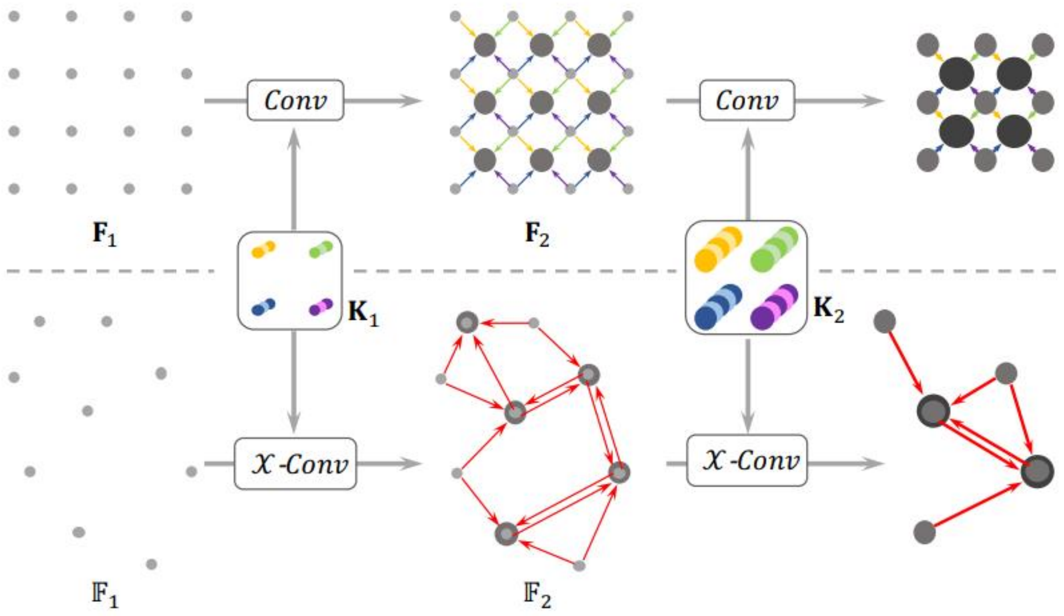


PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

PointCNN, 2018

$\mathcal{X}$-Convolution

- weighting of the input features
- permutation of the points



ALGORITHM 1: $\mathcal{X}$-Conv Operator

Input : $\mathbf{K}, p, \mathbf{P}, \mathbf{F}$

Output : $\mathbf{F}_p$ ▷ Features "projected", or "aggregated", into representative point p

1: $\mathbf{P}' \leftarrow \mathbf{P} - p$ ▷ Move $\mathbf{P}$ to local coordinate system of p

2: $\mathbf{F}_\delta \leftarrow MLP_\delta(\mathbf{P}')$ ▷ **Individually** lift each point into C_δ dimensional space

3: $\mathbf{F}_* \leftarrow [\mathbf{F}_\delta, \mathbf{F}]$ ▷ Concatenate $\mathbf{F}_\delta$ and $\mathbf{F}$, $\mathbf{F}_*$ is a $K \times (C_\delta + C_1)$ matrix

4: $\mathcal{X} \leftarrow MLP(\mathbf{P}')$ ▷ Learn the $K \times K$ $\mathcal{X}$-transformation matrix

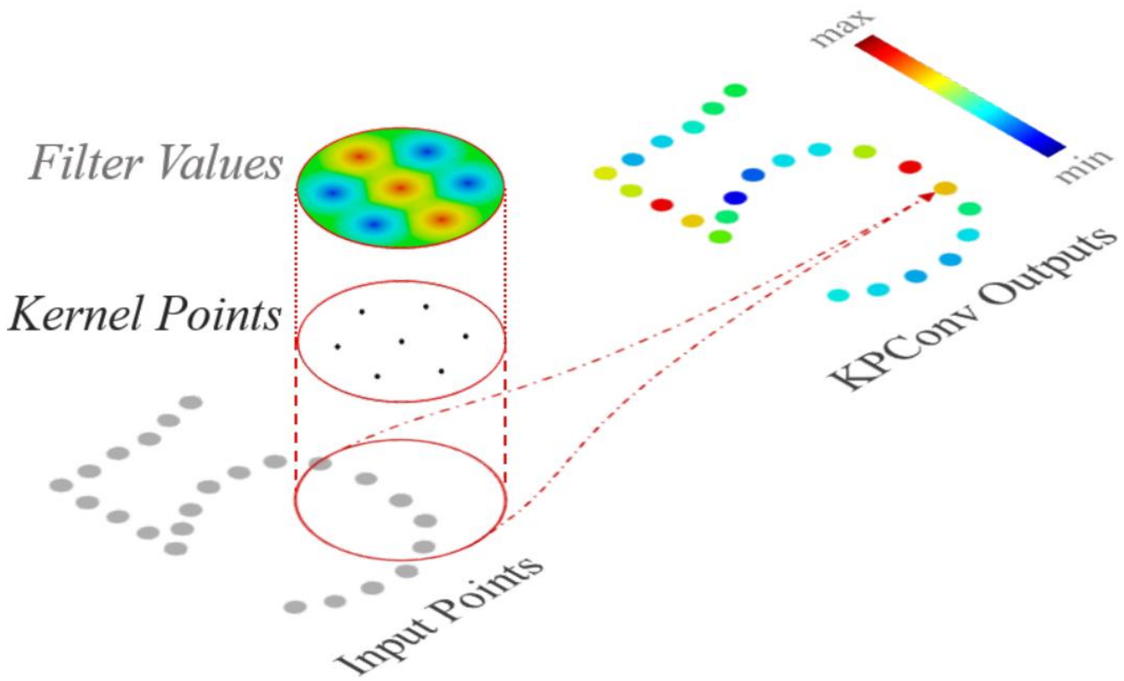
5: $\mathbf{F}_\mathcal{X} \leftarrow \mathcal{X} \times \mathbf{F}_*$ ▷ Weight and permute $\mathbf{F}_*$ with the learnt $\mathcal{X}$

6: $\mathbf{F}_p \leftarrow \mathrm{Conv}(\mathbf{K}, \mathbf{F}_\mathcal{X})$ ▷ Finally, typical convolution between $\mathbf{K}$ and $\mathbf{F}_\mathcal{X}$

PointCNN: Convolution On X -Transformed Points

KPConv, 2019

KPConv use any number of kernel points with learned continuous location to form the convolution operator.

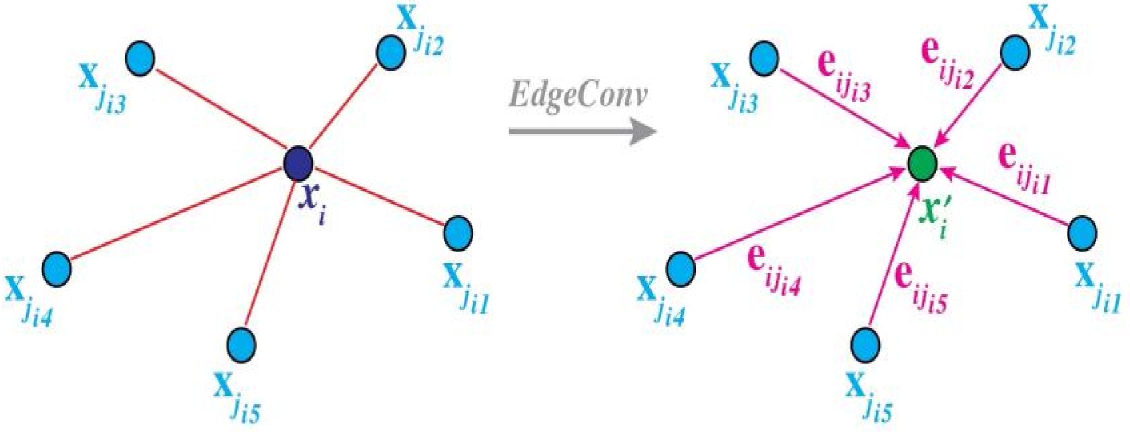


$$g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k$$

KPConv: Flexible and Deformable Convolution for Point Clouds

DGCNN, 2018

- Dynamically constructed graph
- EdgeConv incorporates local neighborhood information



$$e'_{ijm} = \mathrm{ReLU}(\boldsymbol{\theta}_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \boldsymbol{\phi}_m \cdot \mathbf{x}_i),$$

which can be implemented as a shared MLP, and taking

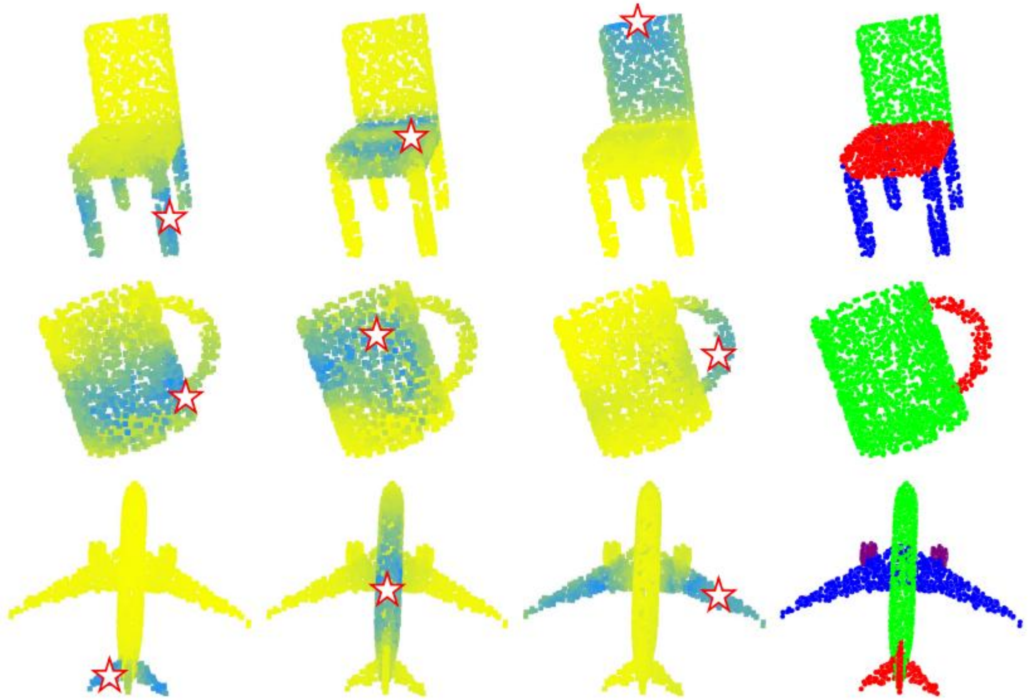
$$x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm},$$

where $\Theta = (\theta_1, \ldots, \theta_M, \phi_1, \ldots, \phi_M)$

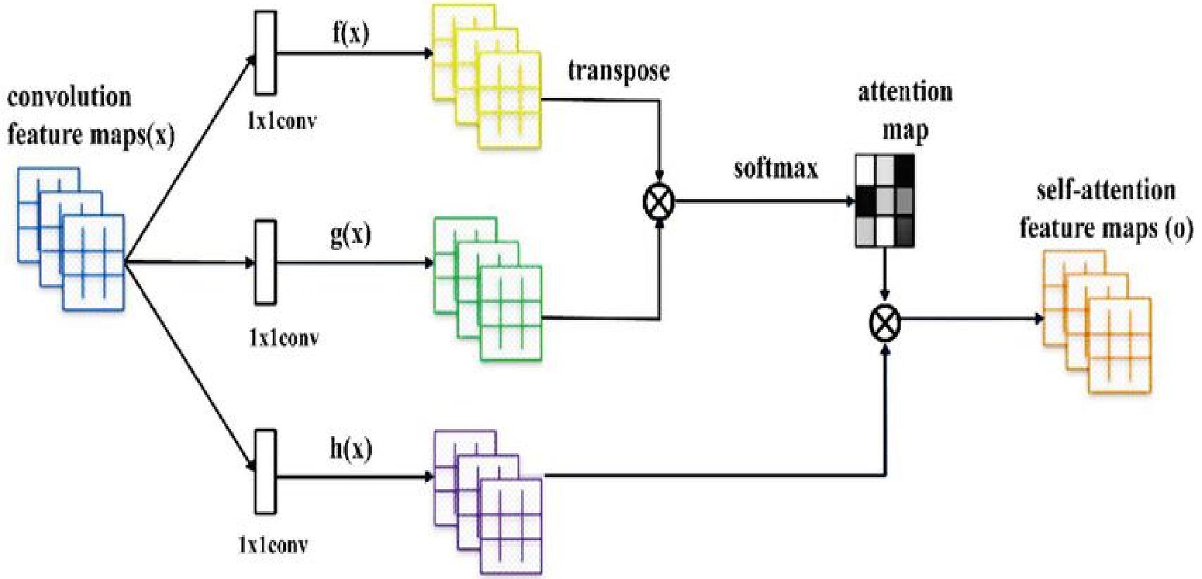
Dynamic Graph CNN for Learning on Point Clouds

PCT, 2019

PCT uses the attention mechanism to learn the aggregation of unordered point features.



Point-wise attention map for different query points indicated by ☆



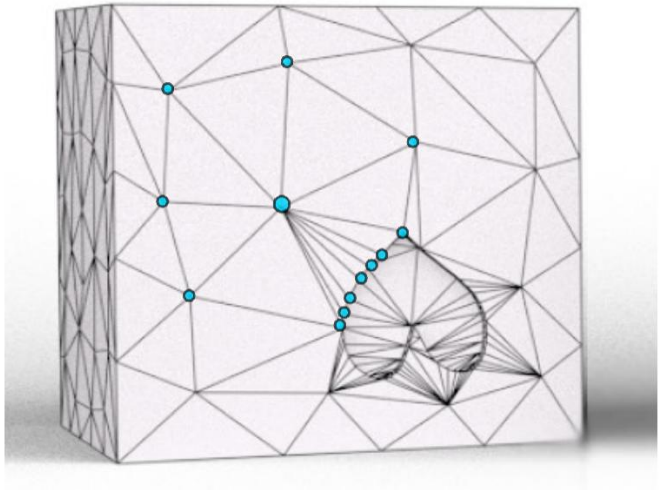
Self-Attention Mechanism

PCT: Point Cloud Transformer

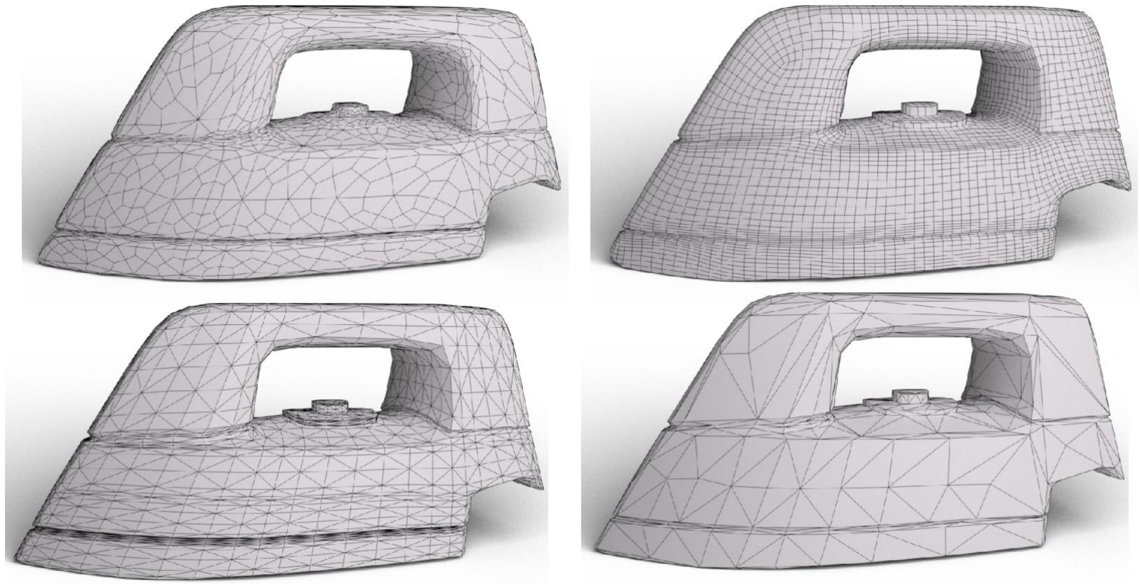
Geometric Deep Learning for 3D Meshes

三维网格数据

Meshes face similar challenges, e.g. irregular, unordered, inconsistent issue.



Irregular and unordered



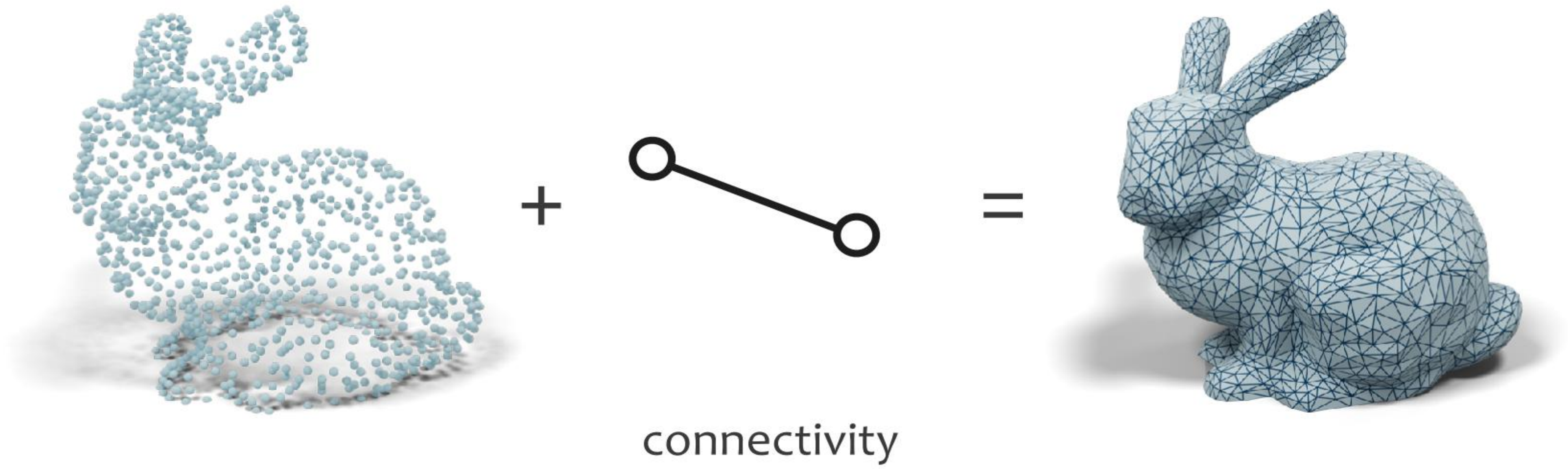
Inconsistent

An Introduction to Deep Learning on Meshes

三维网格数据

Meshes face similar challenges, e.g. irregular, unordered, inconsistent issue.

But they are different from point clouds or general graphs.

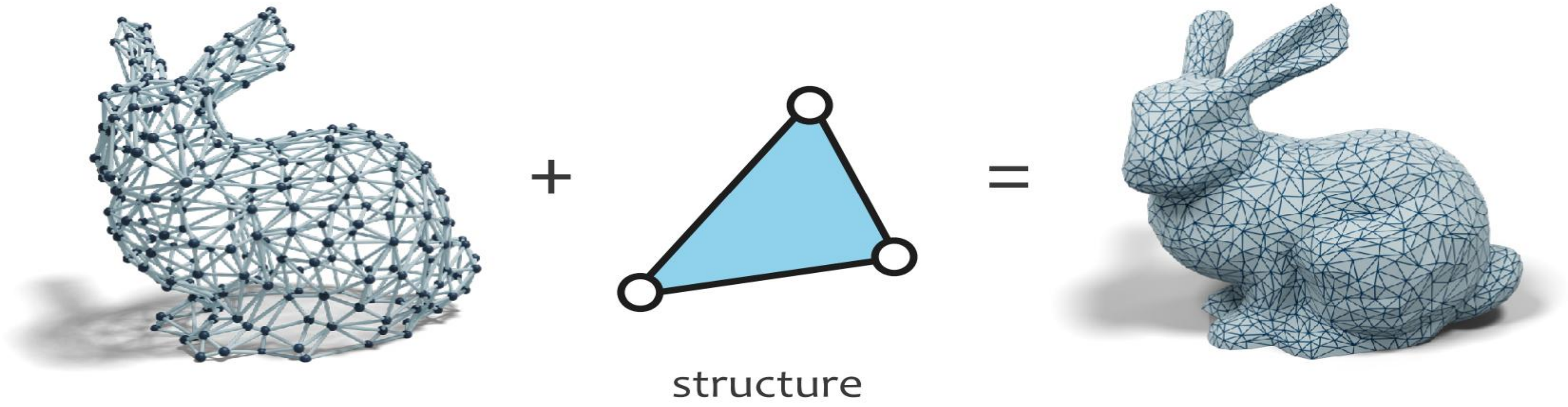


An Introduction to Deep Learning on Meshes

三维网格数据

Meshes face similar challenges, e.g. irregular, unordered, inconsistent issue.

But they are different from point clouds or general graphs.



An Introduction to Deep Learning on Meshes

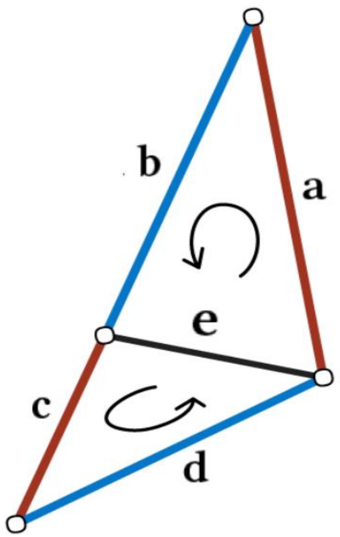
MeshCNN, 2019

Mesh edges are analogized to pixels of images.

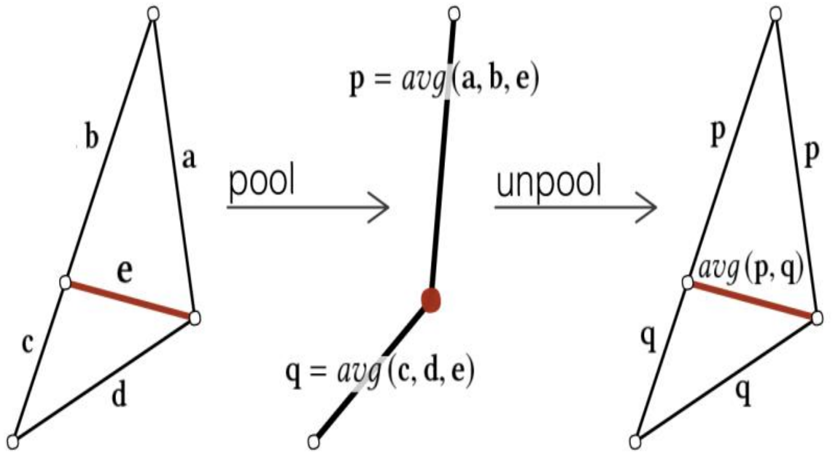
- Convolutions are applied on edges and the four edges of incident triangles
- Pooling is applied via an edge collapse operation

Convolution

The 1-ring neighbors of e can be ordered as (a,b,c,d) or (c,d, a,b). So we can aggregate them into two pairs of edges (e.g., a and c, and b and d), and apply simple symmetric functions on each pair (e.g., sum(a,c)).



Convolution



Pooling

Pooling

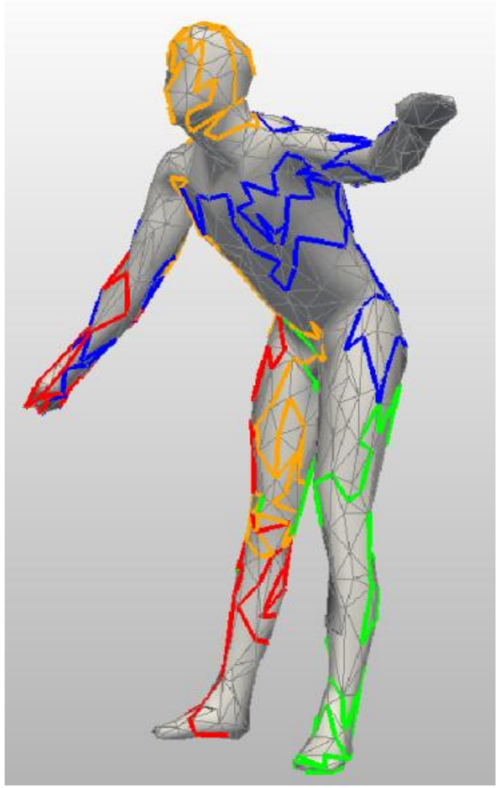
The original 5 edges is transformed into 2 edges after the collapse. We can pool each three edges to one updated edge feature.

MeshCNN: A Network with an Edge

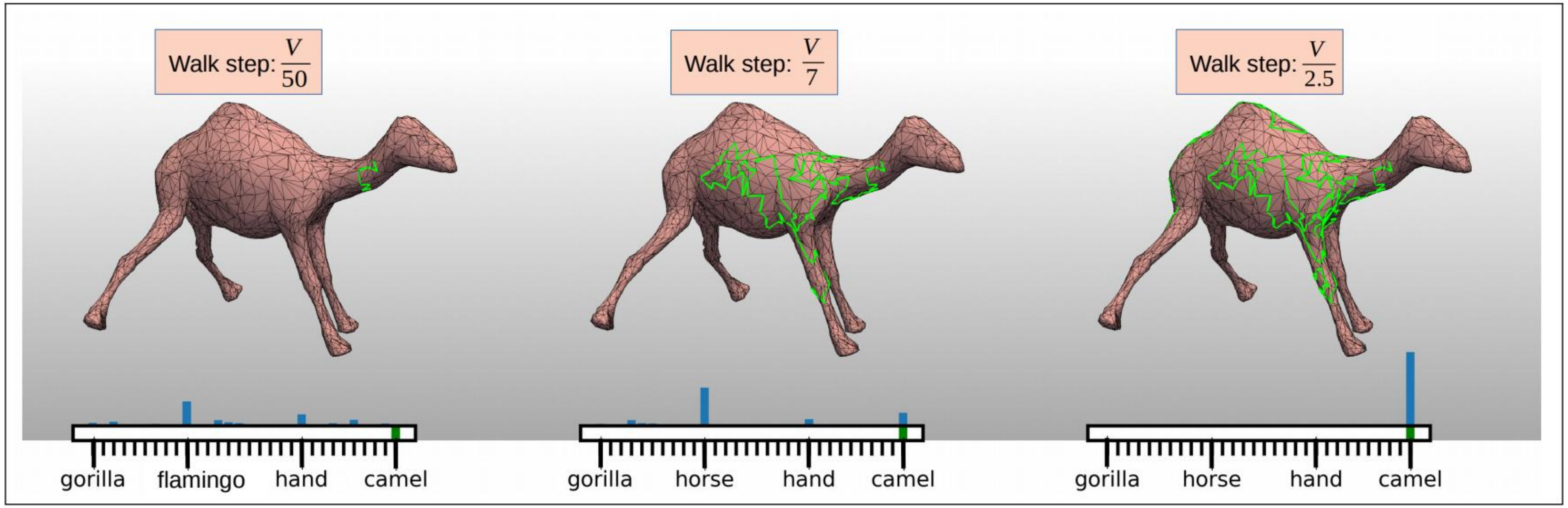
MeshWalker, 2020

Random walk to explore the local and global geometry information.

RNN to aggregate the information along each walk.



(a) 5 walks on the surface

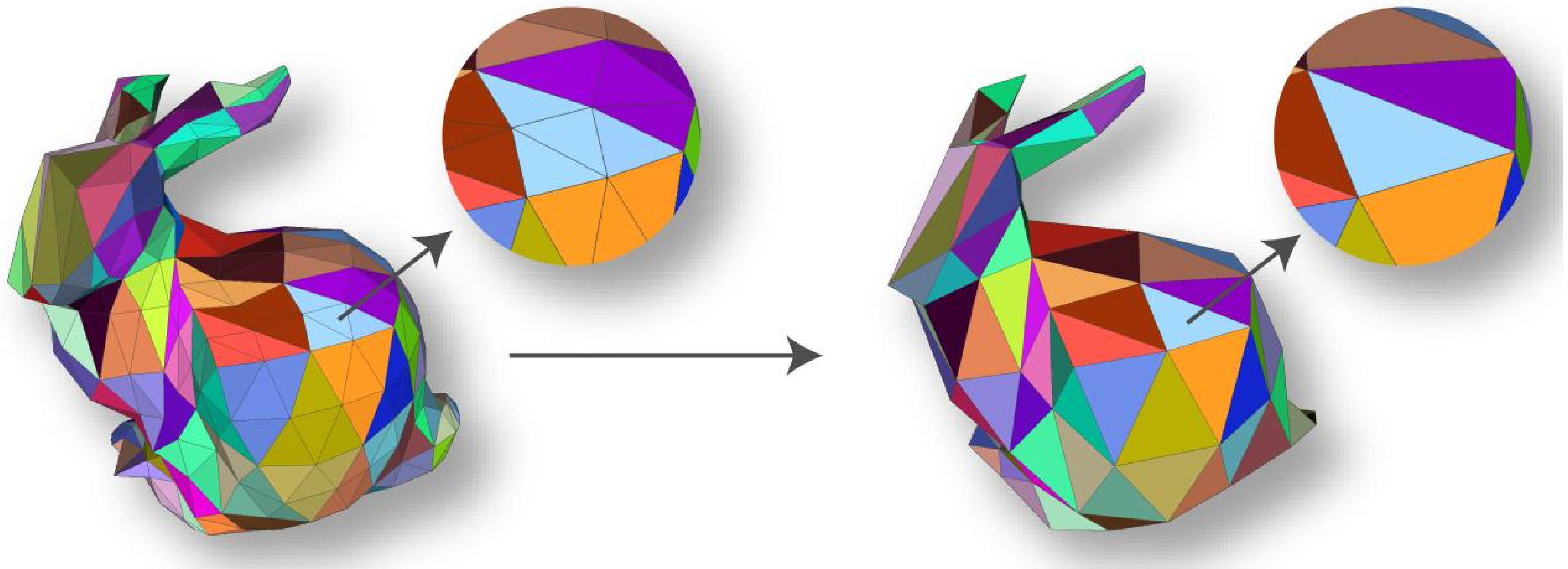


MeshWalker: Deep Mesh Understanding by Random Walks

SubdivNet, 2022

Loop subdivision to construct a hierarchical subdivision structure.

Regular convolution and pooling on the hierarchical structure.



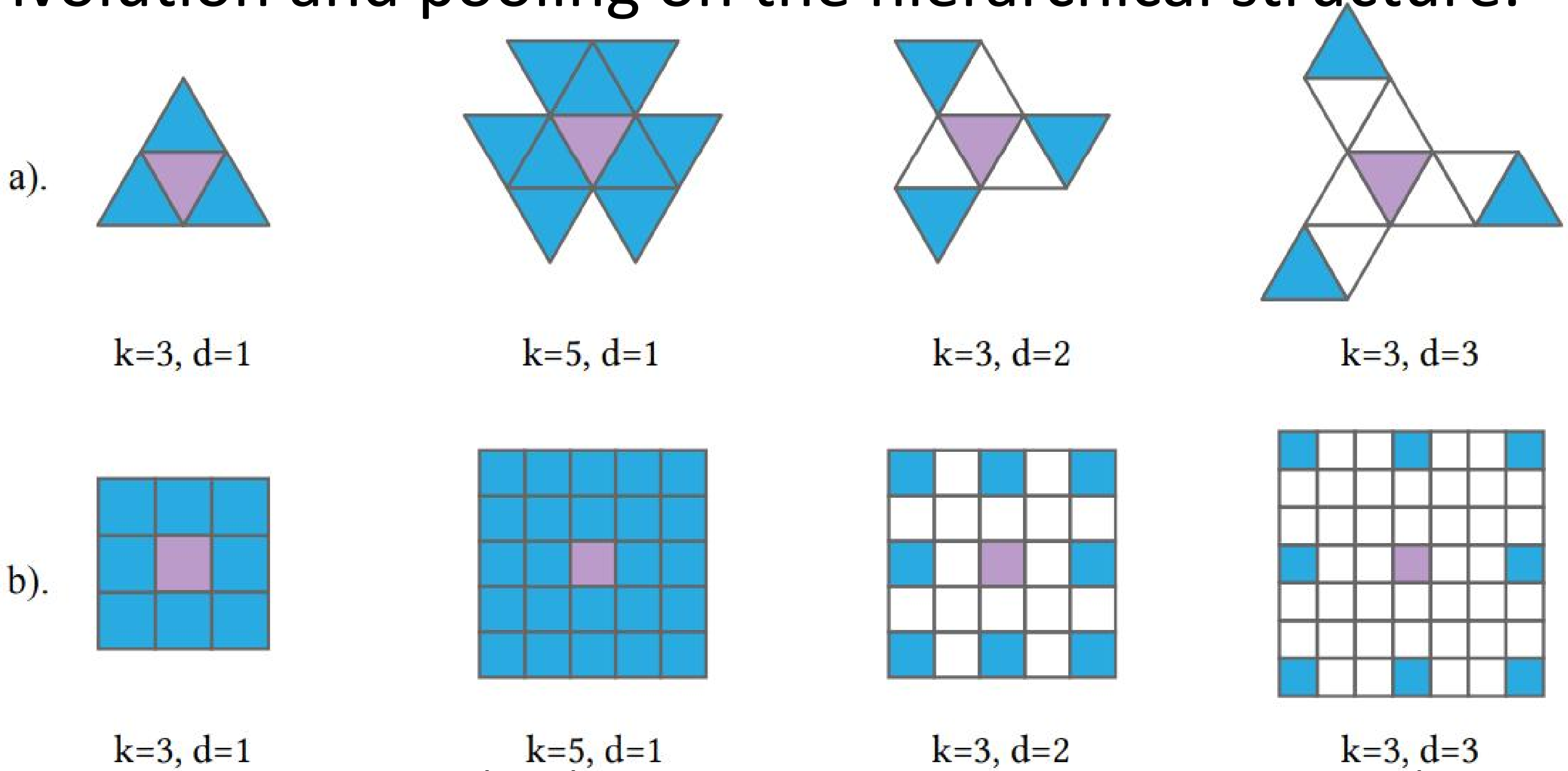
Loop Subdivision

Subdivision-Based Mesh Convolution Networks

SubdivNet, 2022

Loop subdivision to construct a hierarchical subdivision structure.

Regular convolution and pooling on the hierarchical structure.



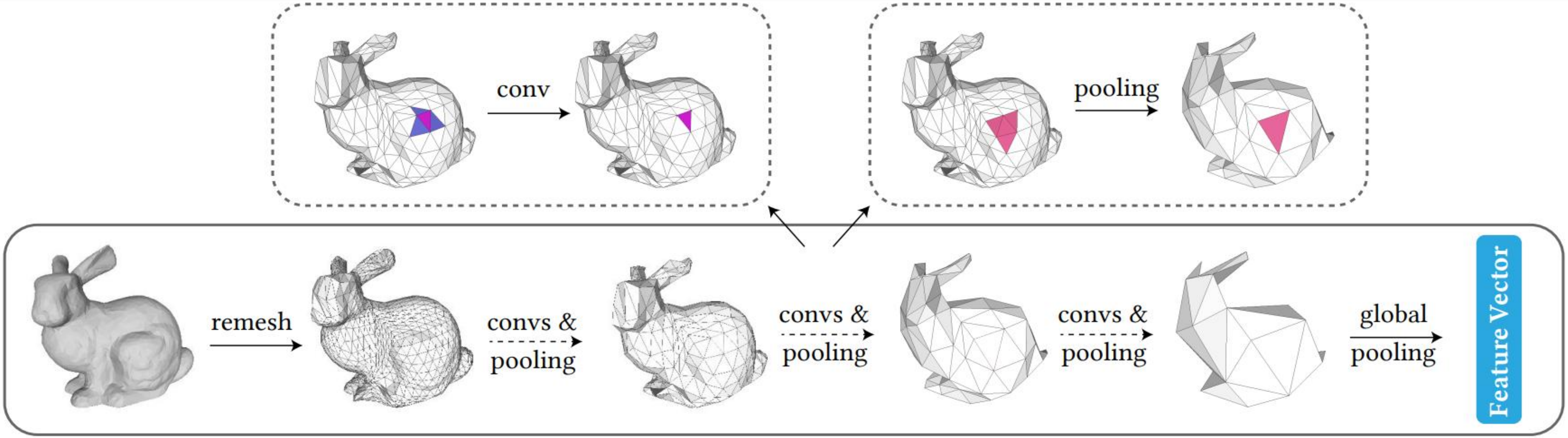
Mesh convolution (Top) and the analogous lattice convolution (Bottom)

Subdivision-Based Mesh Convolution Networks

SubdivNet, 2022

Loop subdivision to construct a hierarchical subdivision structure.

Regular convolution and pooling on the hierarchical structure.

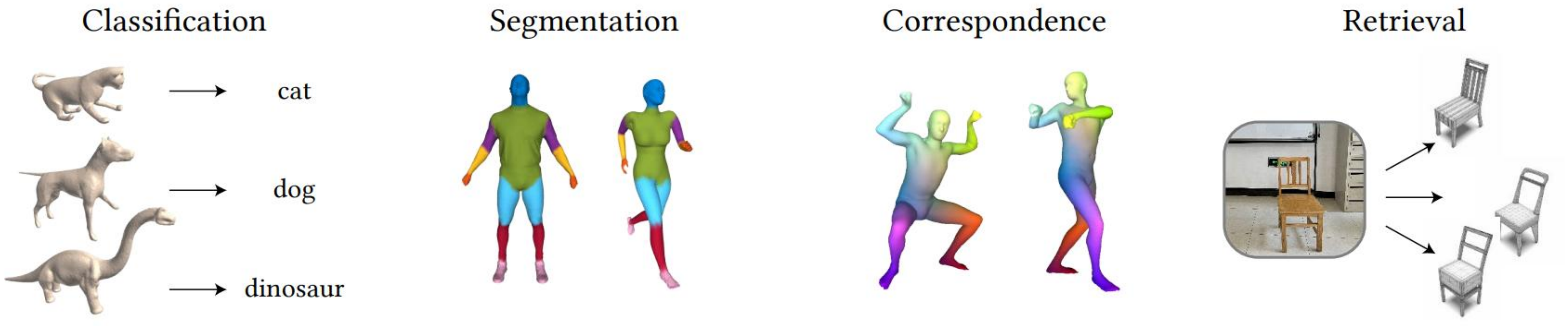


Subdivision-Based Mesh Convolution Networks

What can we do with the help of these backbone networks?

几何处理任务

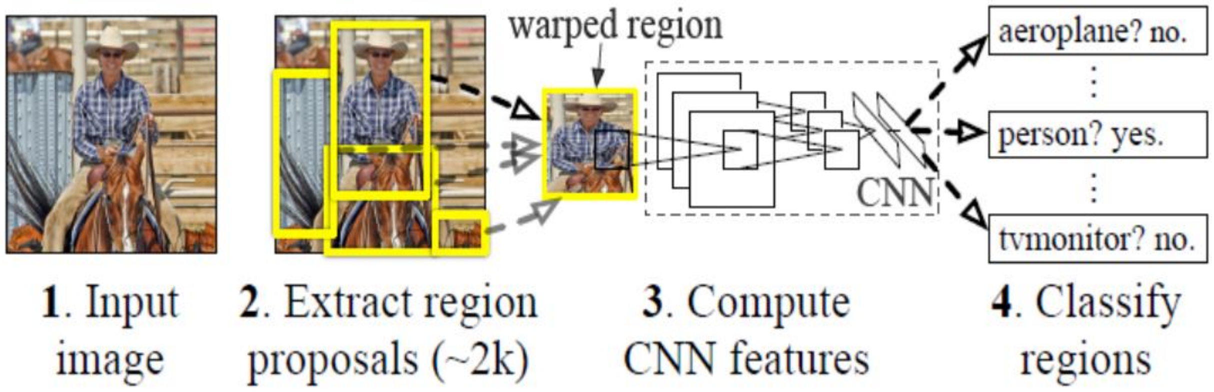
- We can classify the shape feature and point-wise features respectively for global recognition and local analysis tasks.



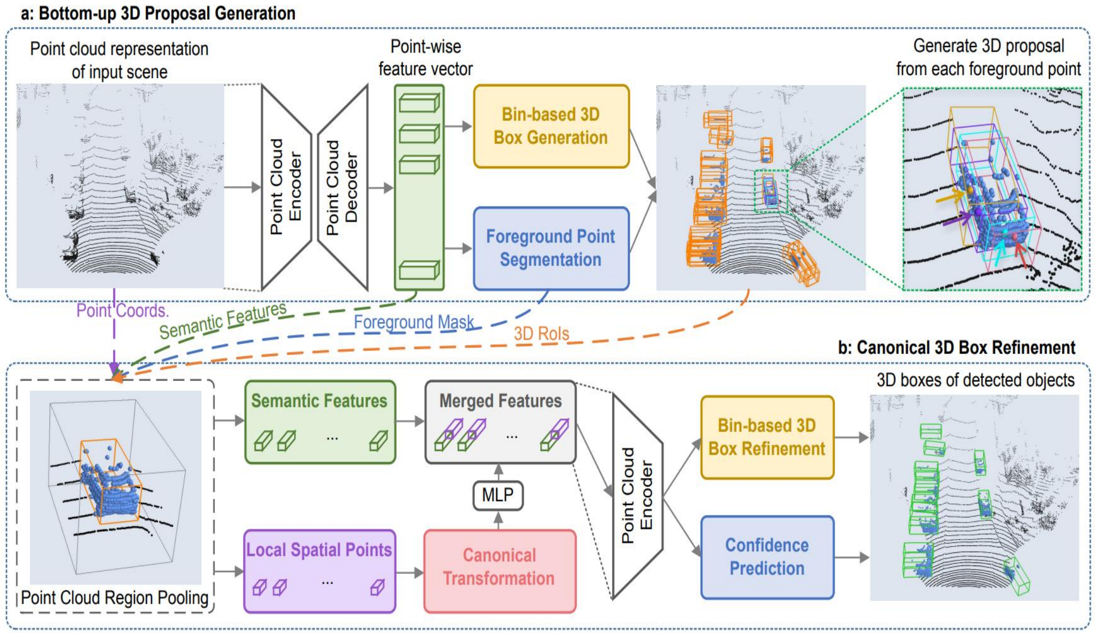
global　　local　　local　　global

几何处理任务

- Region Proposal Methods for 3D object detection / instance segmentation.



Region Proposal

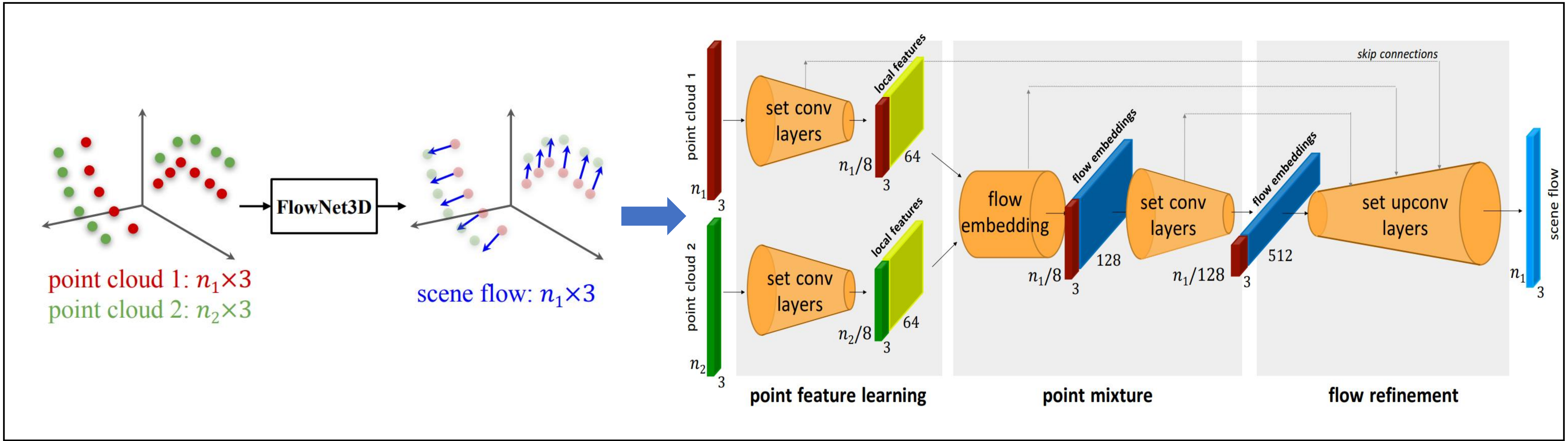


PointRCNN

Directly replace the backbone network from 2D CNN to 3D networks

几何处理任务

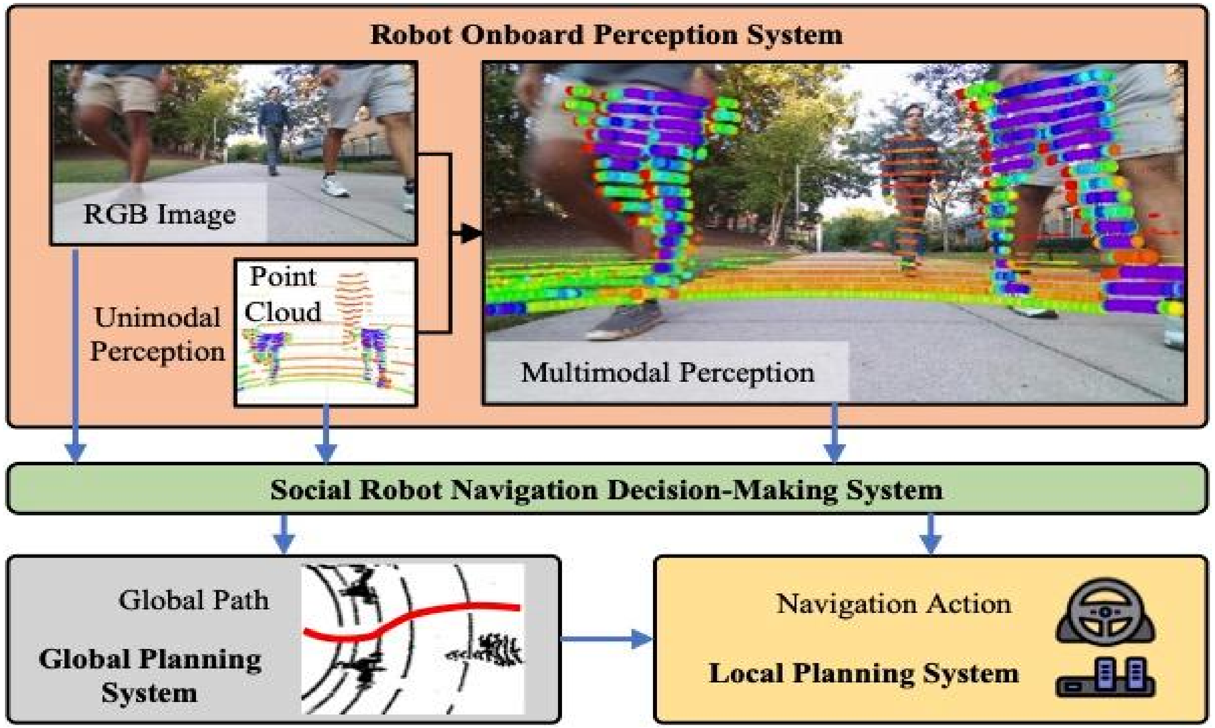
- Design the input and output data format for 3D object tracking / motion prediction.



Flexibly utilize the geometric layers for specific processing

具体任务具体分析

◆Multi-modal Perception in practice

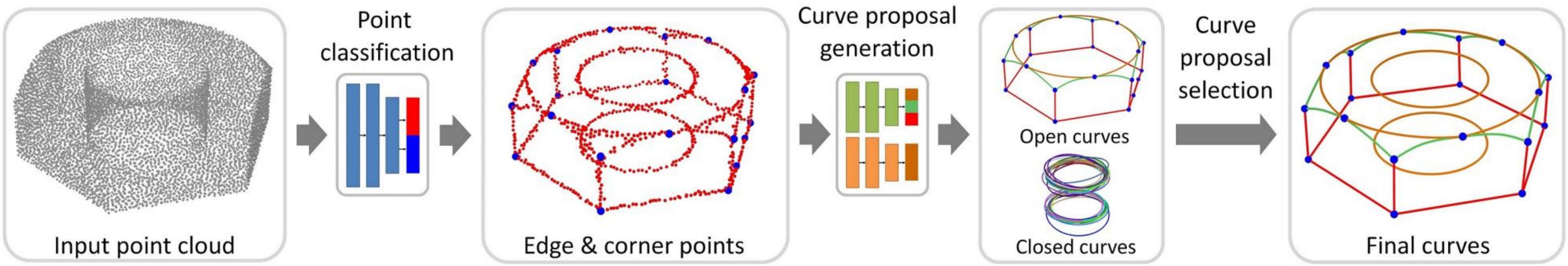


RGB + point cloud

Interaction-based Reconstruction

具体任务具体分析

◆A sequence of networks together to complete a complex task.



PIE-NET

Summary

Summary

- Geometric deep learning study the fundamental network design for non-euclidean data, e.g. various 3D surface representations.
- There has been a vast of backbone networks designed for point clouds and meshes.
- We can flexibly select and combine the network modules for specific tasks.

Thank you