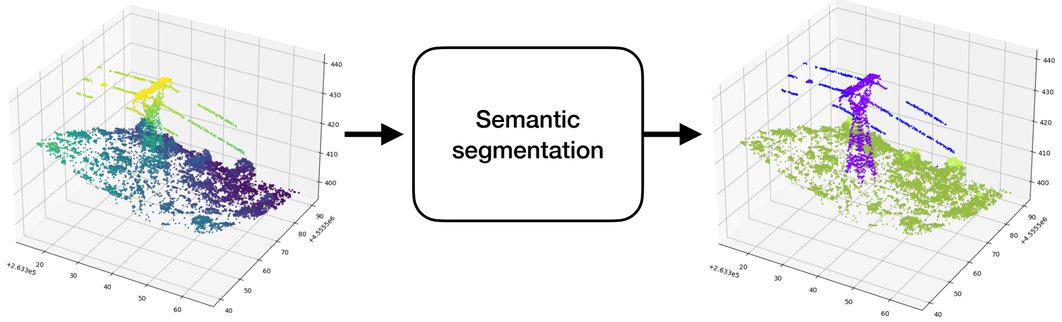
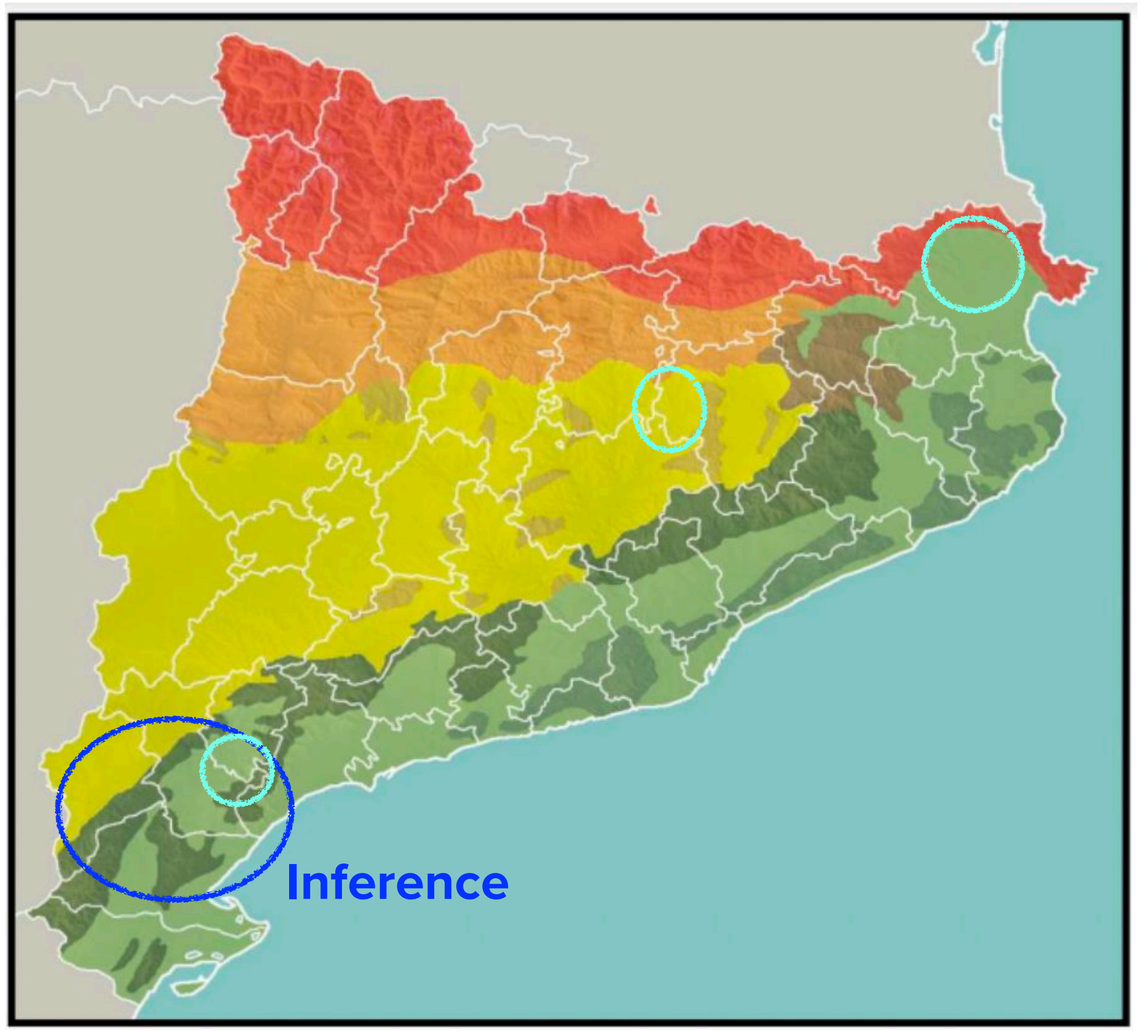
Into the world of unstructured 3D data

Mariona Carós

UNIVERSITAT DE BARCELONA

Industrial PhD





Labeled data

UNIVERSITAT DE BARCELONA

ICGC Institut Cartogràfic i Geològic de Catalunya

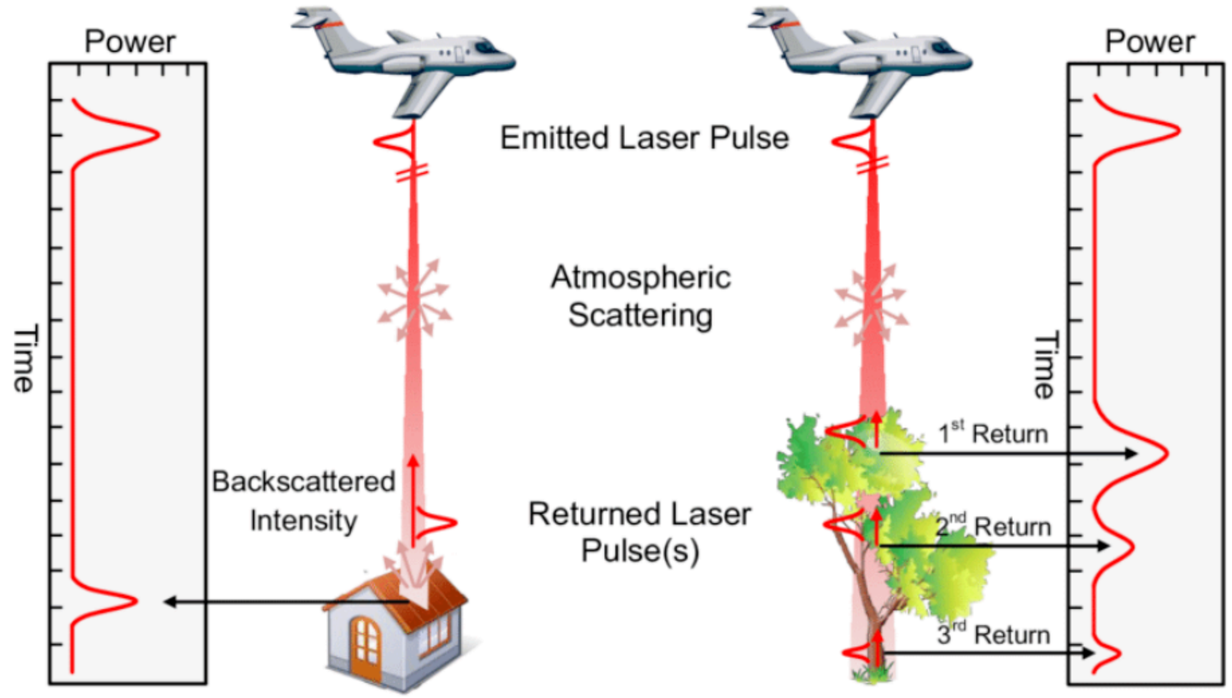
DOCTORATS INDUSTRIALS

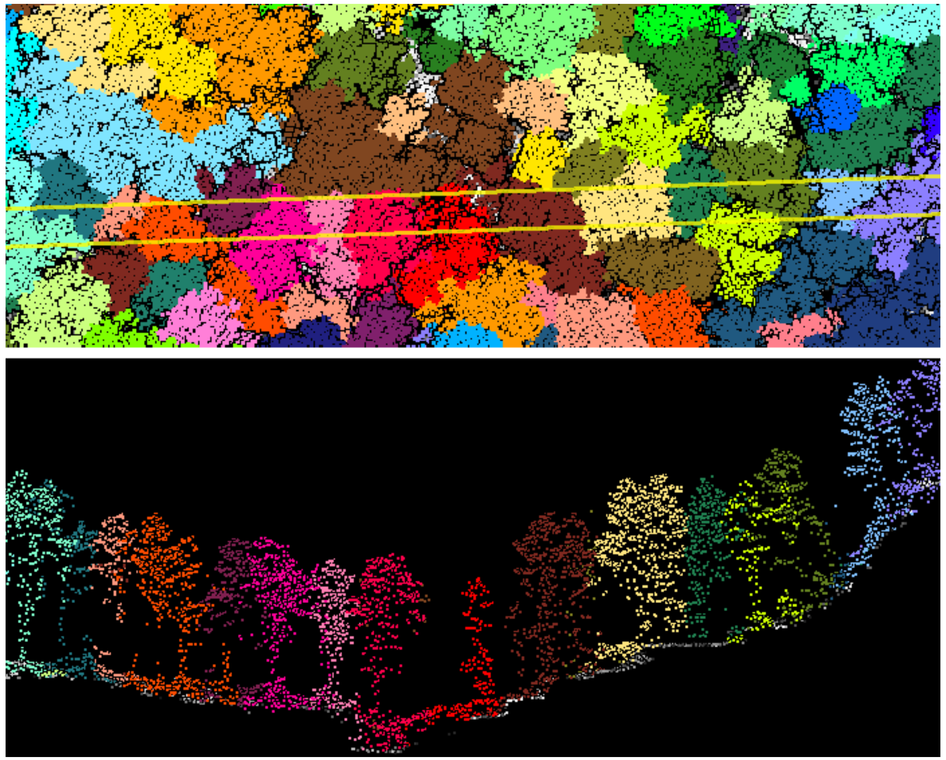
Semantic segmentation of a LiDAR scene

Segmentatin of power lines and wind turbines

LiDAR multi-return capability

- Lidar sensors get multiple returns from a single laser pulse. This capability allows the creation of detailed vertical profiles of the surveyed area and helps in distinguishing between different surfaces and objects.

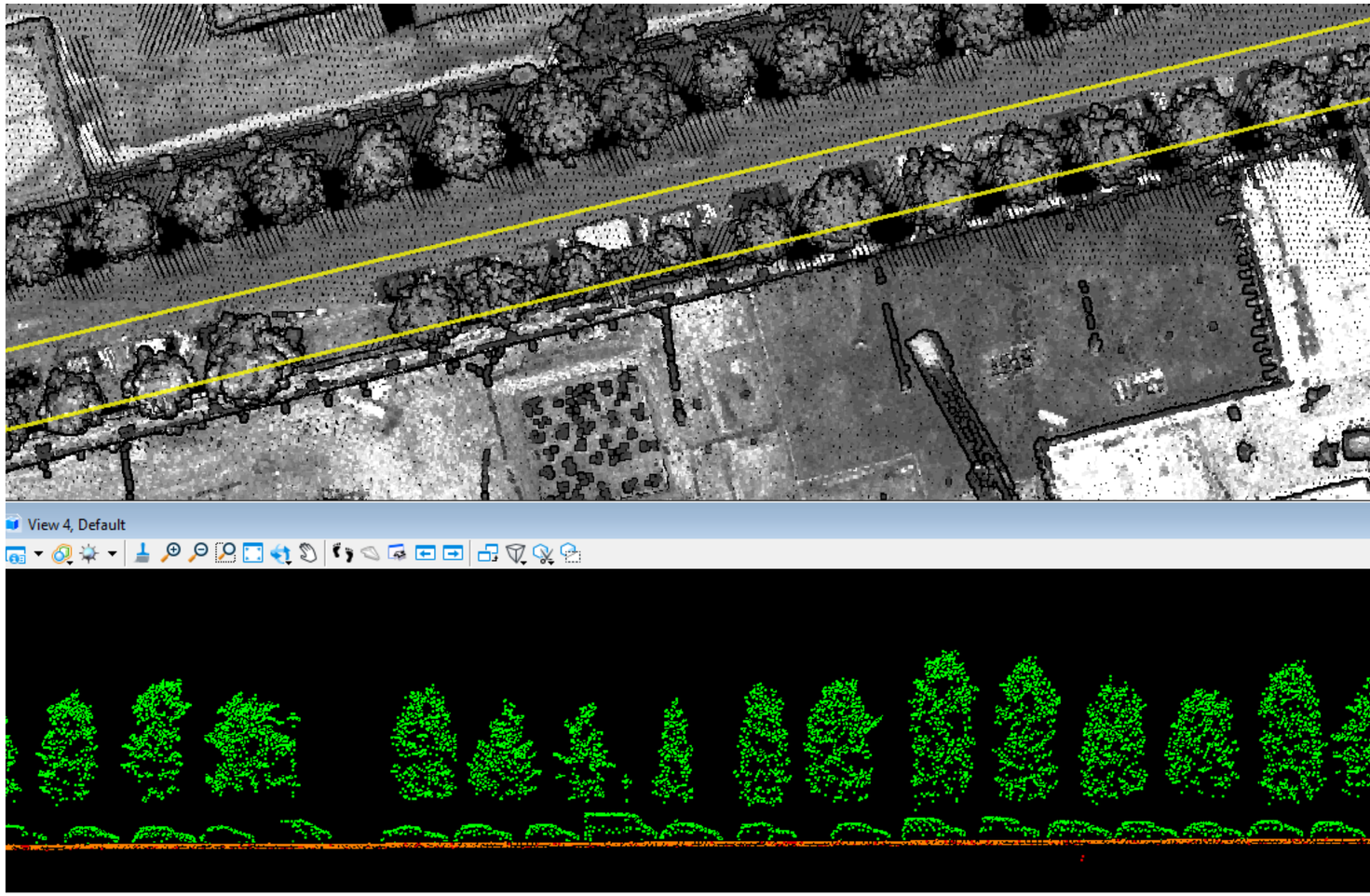




Yan, Wai Yeung & Shaker, Ahmed & El-Ashmawy, Nagwa. (2015). Urban land cover classification using airborne LiDAR data: a review. Remote Sensing of Environment. 158. 295-310. 10.1016/j.rse.2014.11.001.

LiDAR data

- High spatial accuracy
  - Our sensor has a precision of 5cm
- High resolution
  - Each scene have millions of points
- Represented as a point cloud
  - Point features
    - coordinates x, y, z
    - intensity
    - RGB
    - near infrared



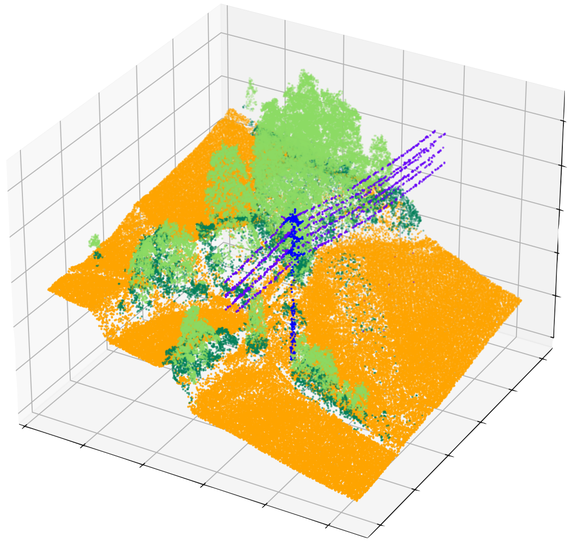
Properties of point sets

- **Invariance under transformations.** The learned representation of the point set should be invariant to rotation, translation or scale.

- **Unordered**. Unlike pixel arrays in images, point cloud is a set of points without specific order.

- **Local structures.** The points are from a space with a distance metric. Points are not isolated, and neighboring points form a meaningful subset.

- The **points are not uniformly distributed in a voxel**. Incorporating these points into a 3D tensor is challenging due to their non-uniform distribution within the voxel.

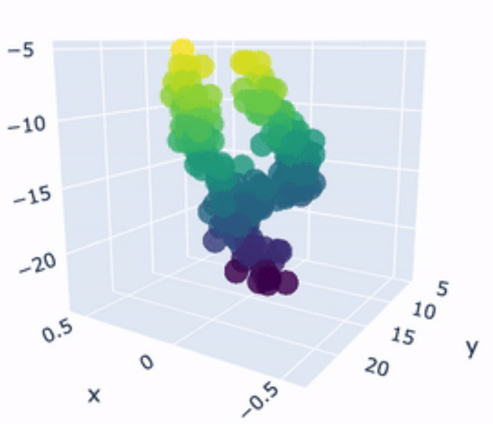


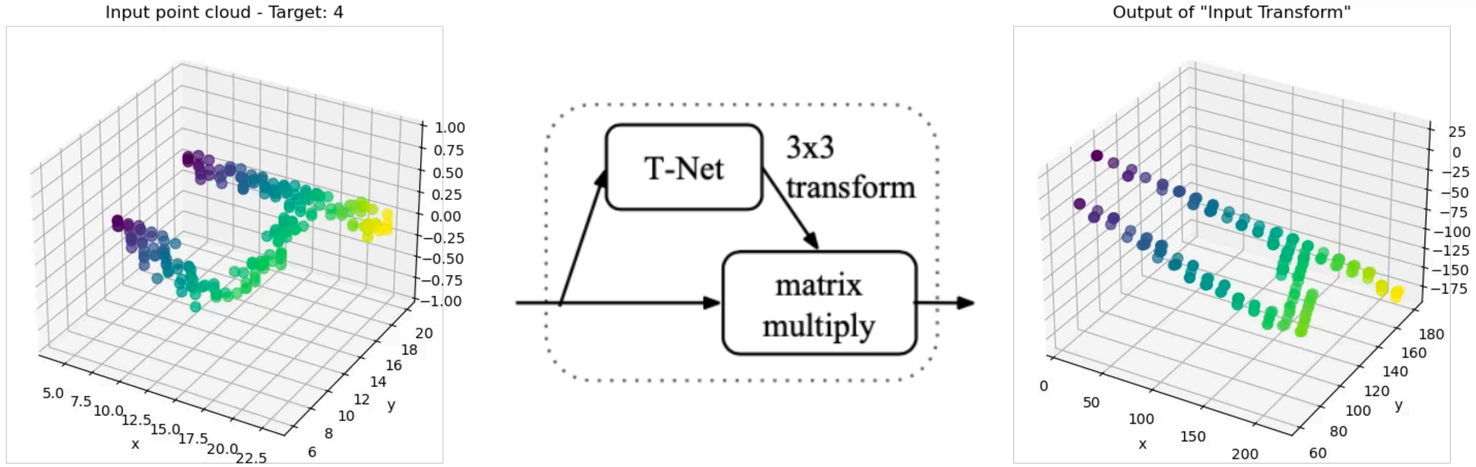
[4000, 8]

PointNet

Invariance under transformations

- *Requirement*: Rotating and translating points all together should not modify the global point cloud category nor the segmentation of the points.
- *Solution*: Transform data into a canonical form by using T-Net

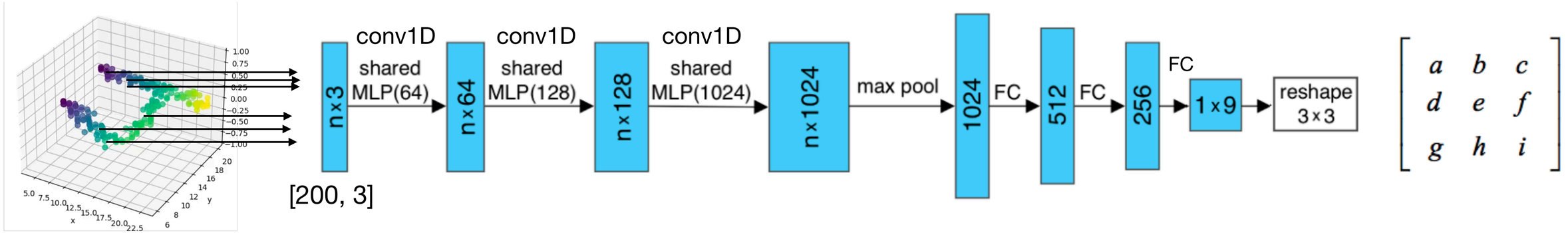




Invariance under transformations

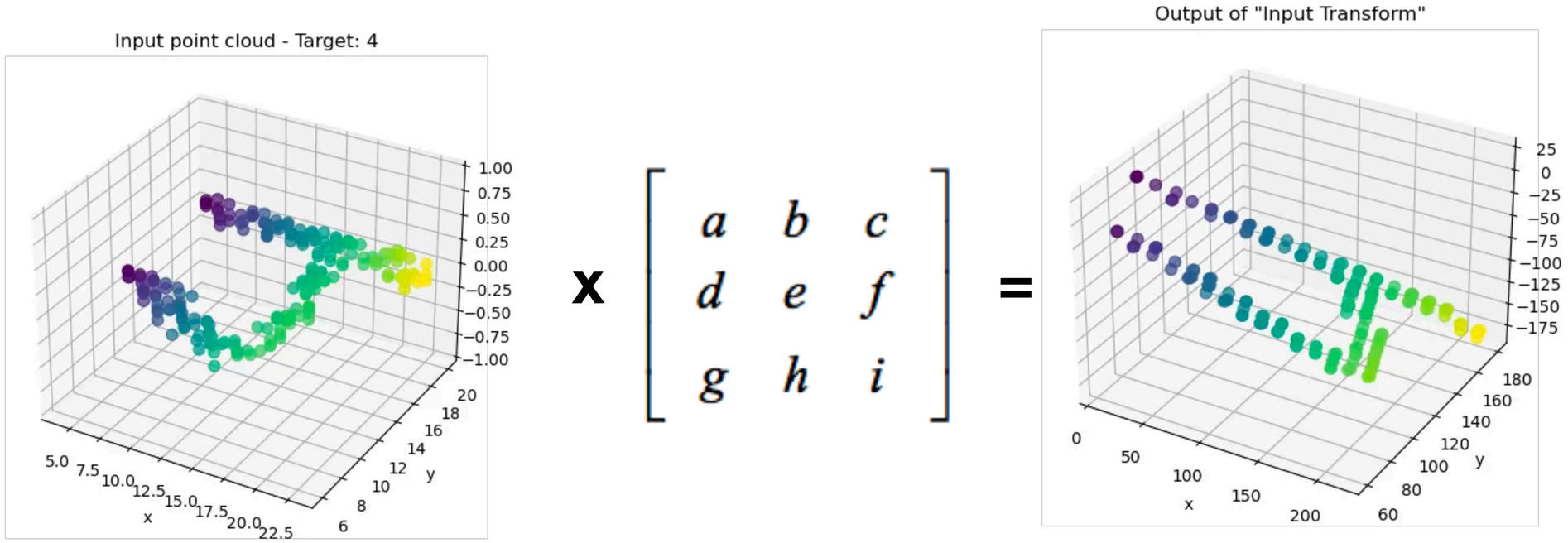
Transformation Net

- How does **T-Net** align points into a canonical form?
- It **learns** an **affine transformation matrix** of 3x3 to be applied to the coordinate of input points (x,y,z)

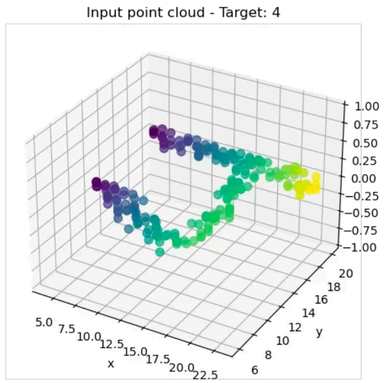


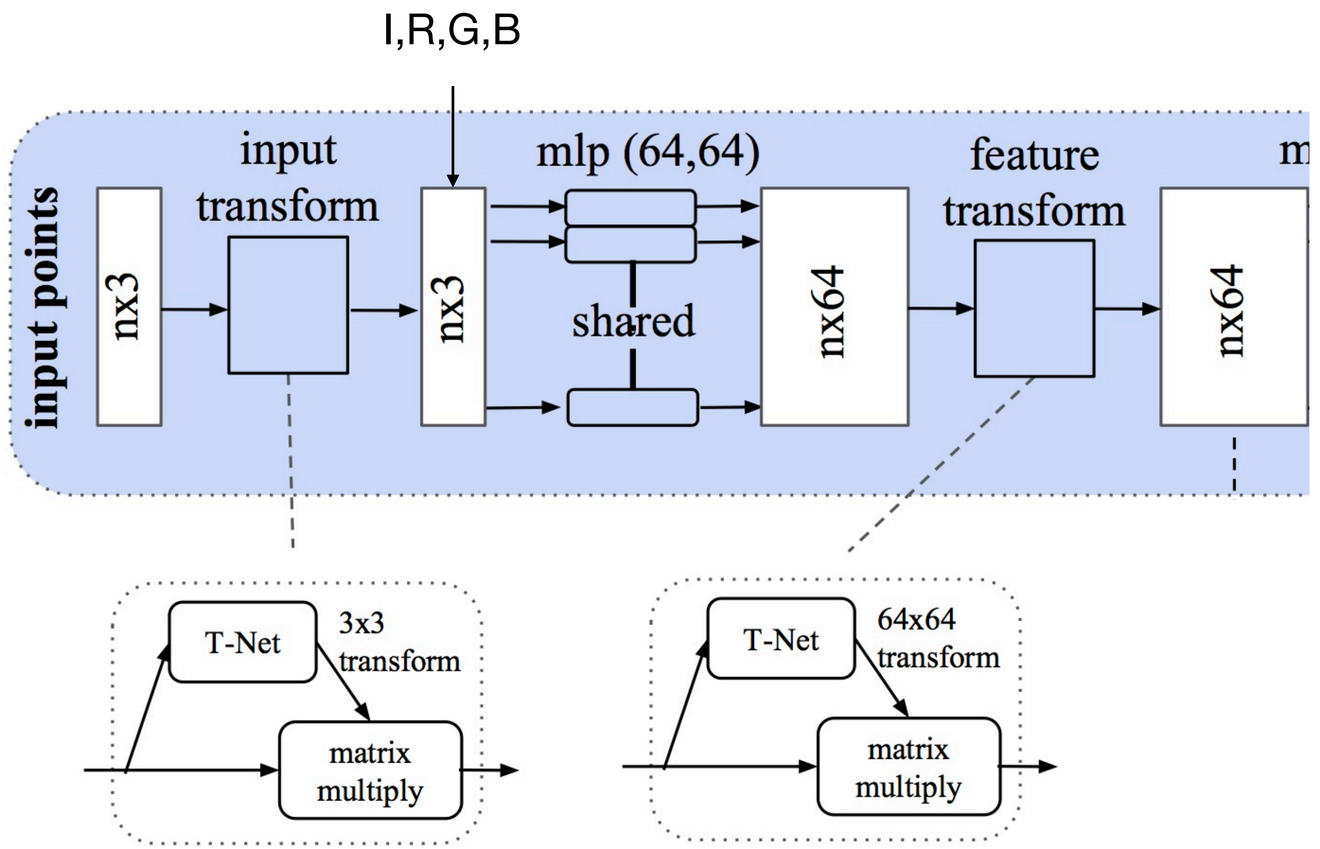
Invariance under transformations

Transformation Net



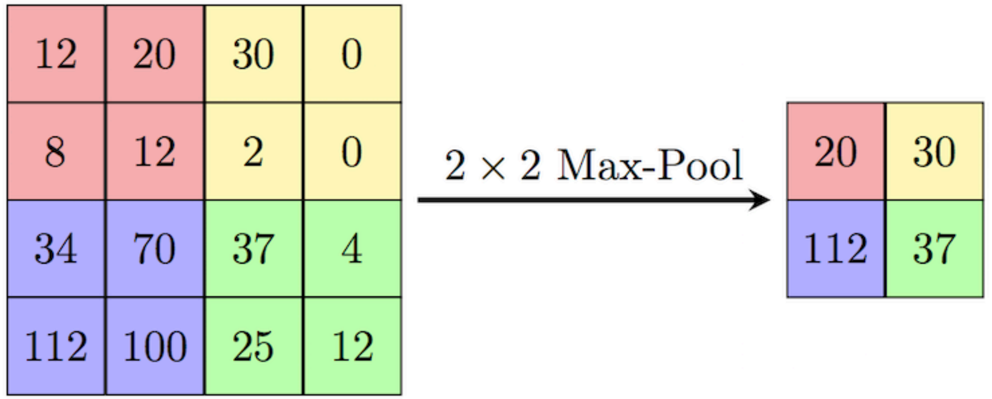
PointNet Architecture



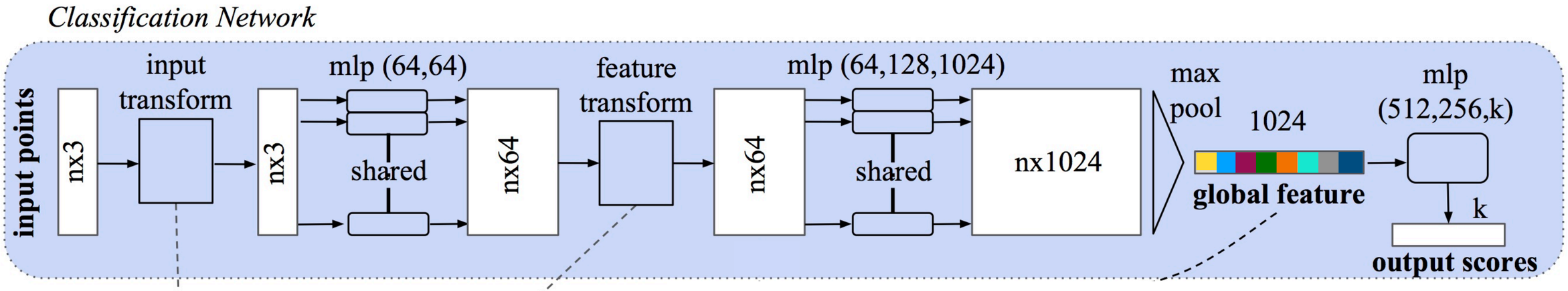


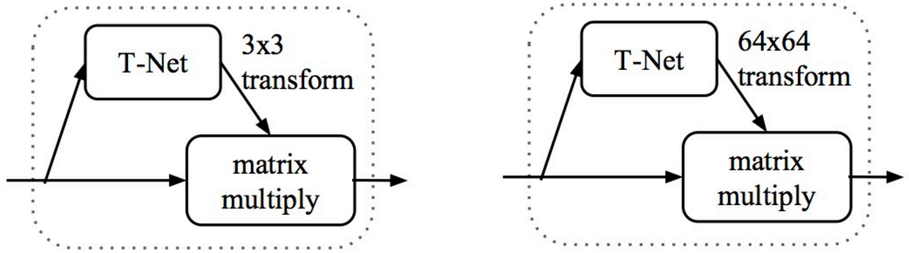
Addressing point permutations

- *Requirement:* The model needs to be invariant to permutations of points
- *Solution*: We turn to the concept of symmetric functions.
  - **Max pooling** as a symmetric function. Like multiplication (*) and addition (+), the order of inputs doesn't alter the result.
    - This property makes max pooling an ideal choice for maintaining invariance to point permutations.

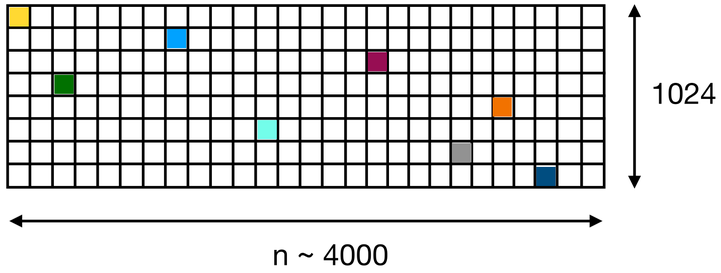


PointNet Architecture

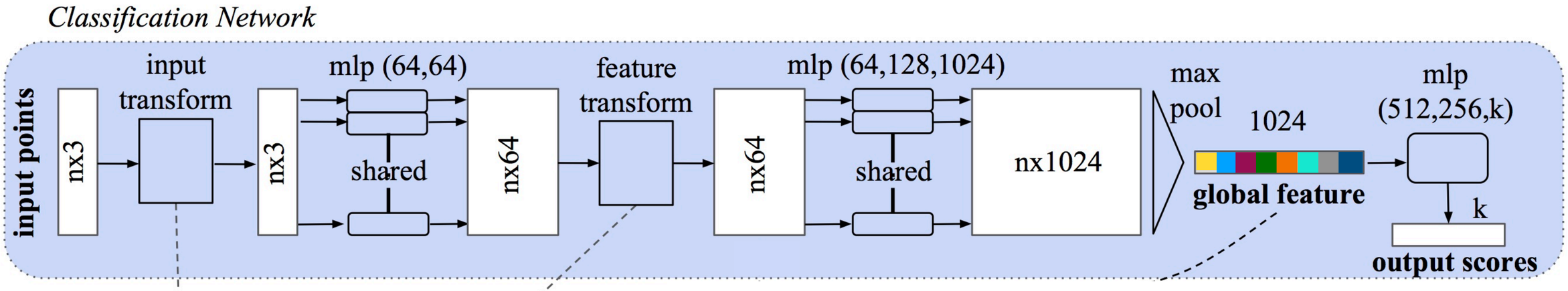


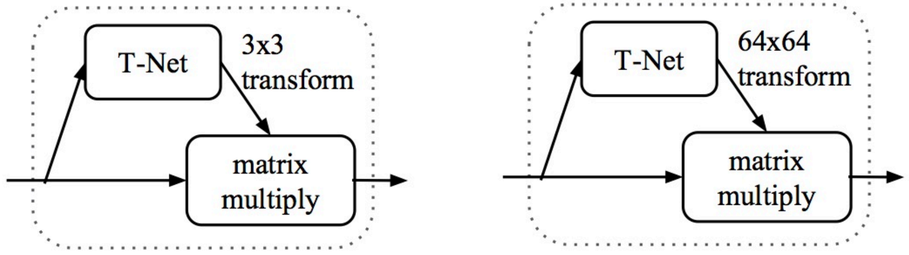


max pooling keeps the maximum of each dimension, generating an **embedding that represents the point cloud**



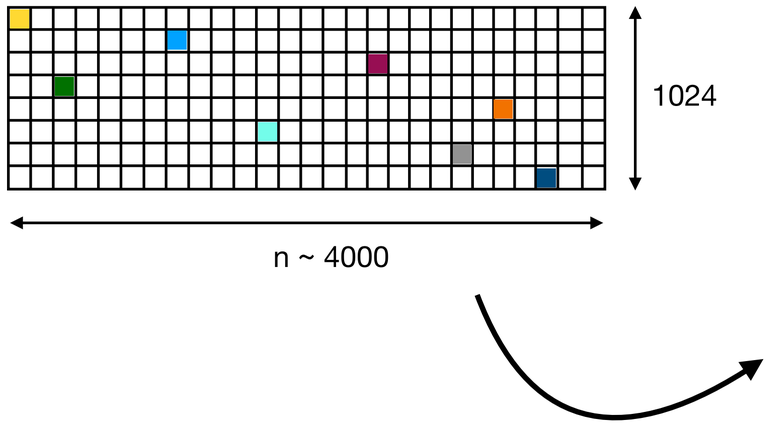
PointNet Architecture

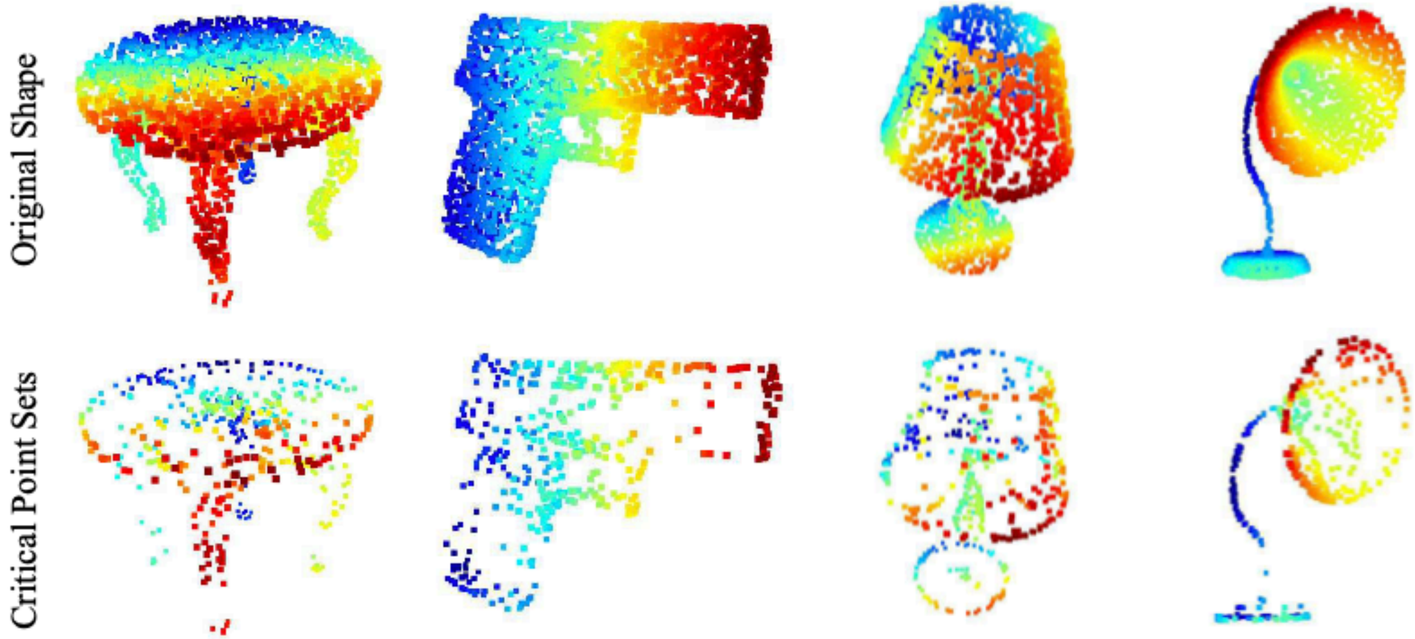




This vector is used for point cloud classification by adding a MLP afterwards

Visualizing PointNet Critical points

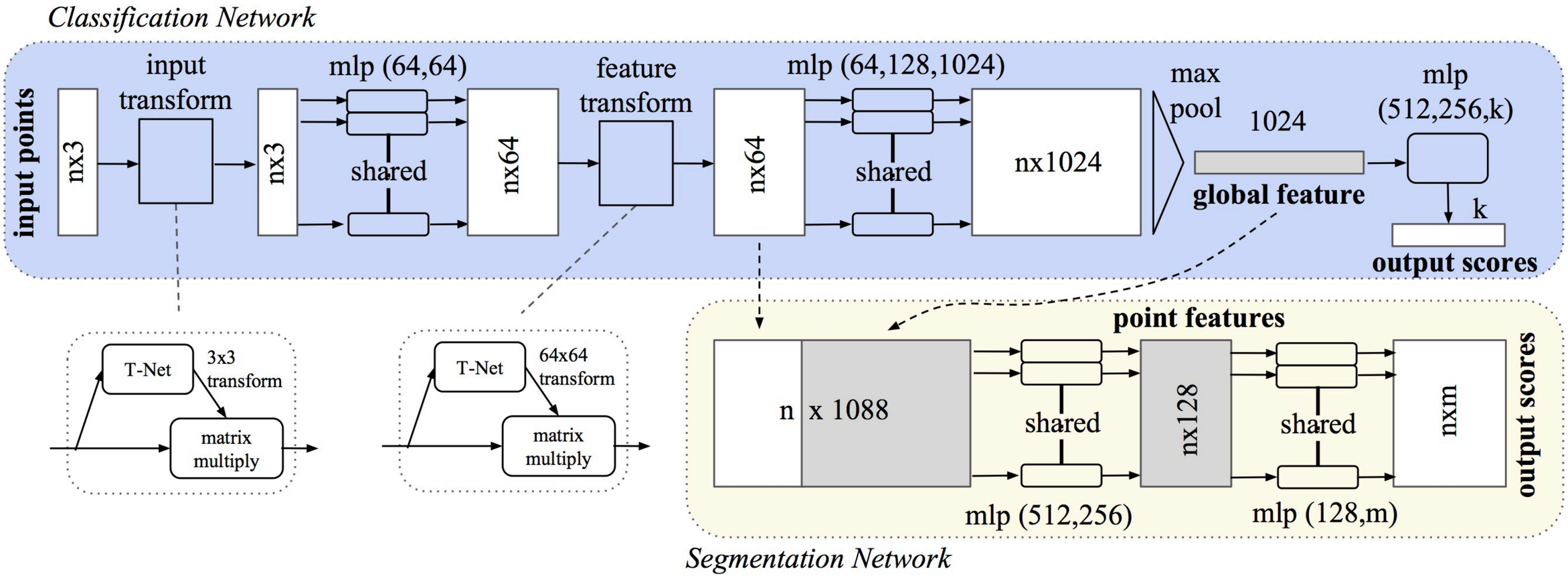




Local structures

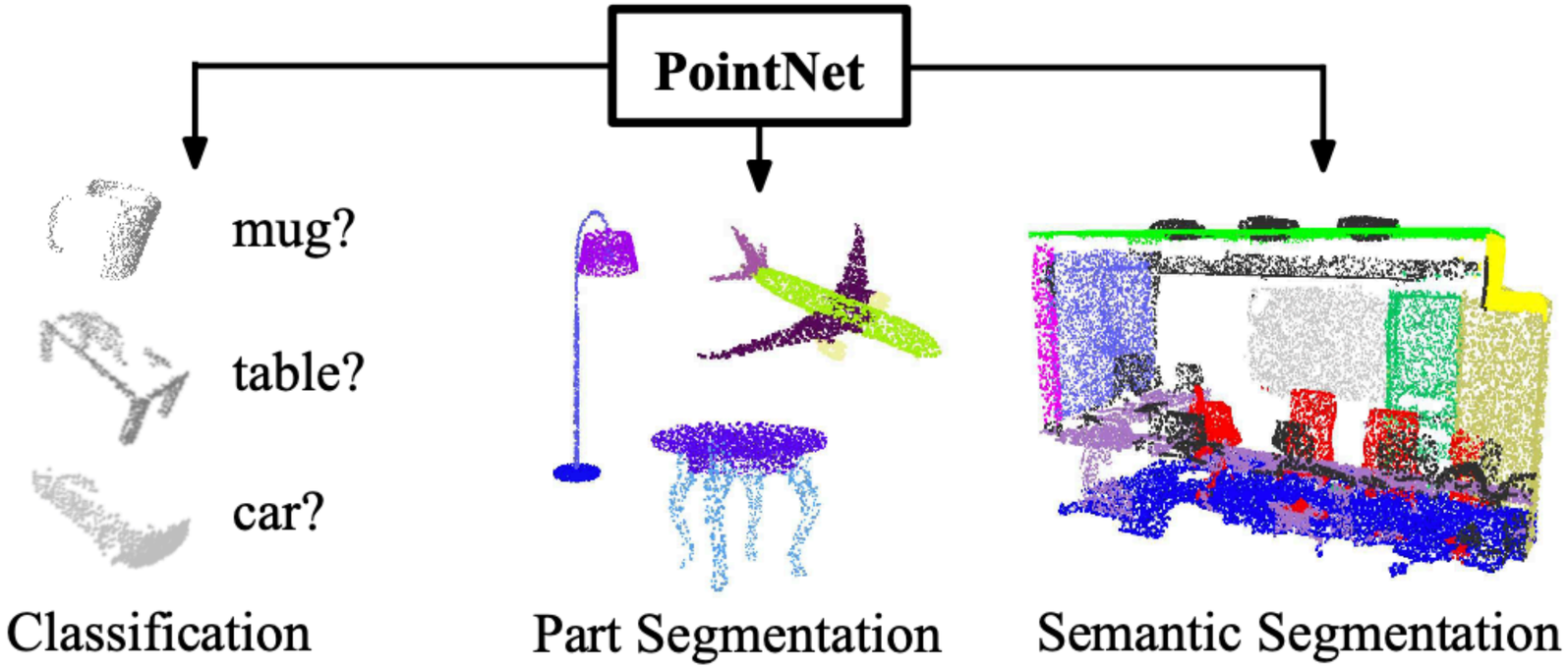
- Points come from a space defined by a distance metric, implying that the proximity of points holds meaningful information. It's not just about individual points; it's about the relationships and interactions among them.
- *Requirement*: Capturing Local Structures
- *Solution*: **Combining Local and Global representations**.
  - Local features zoom in on fine-grained details
  - Global features provide the necessary context for a comprehensive segmentation result.

PointNet Architecture



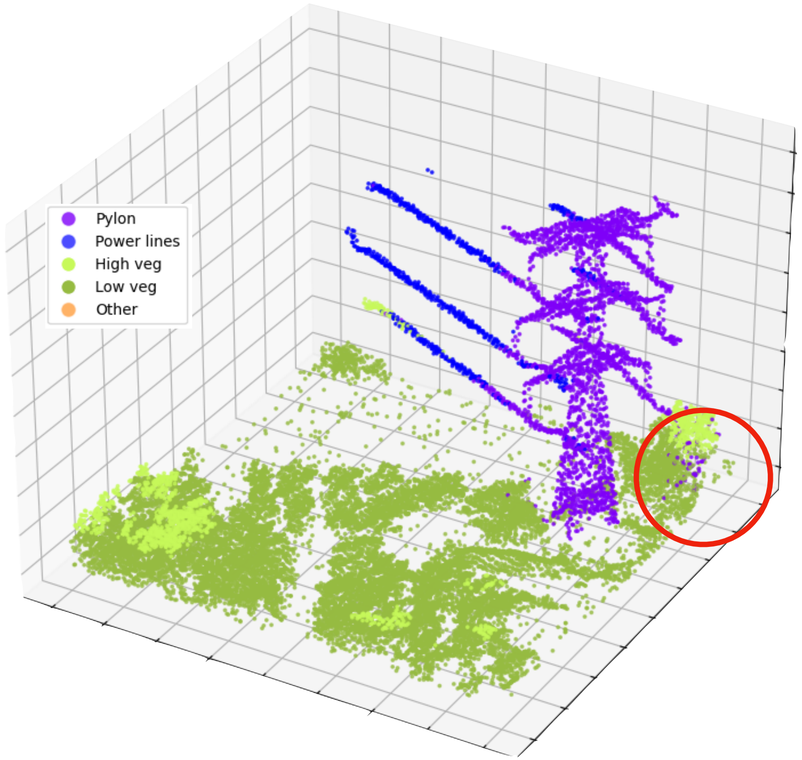
PointNet

Applications



PointNet limitation

Everything seemed to be working, but... There ara some prediction errors due to **local structures not well captured**

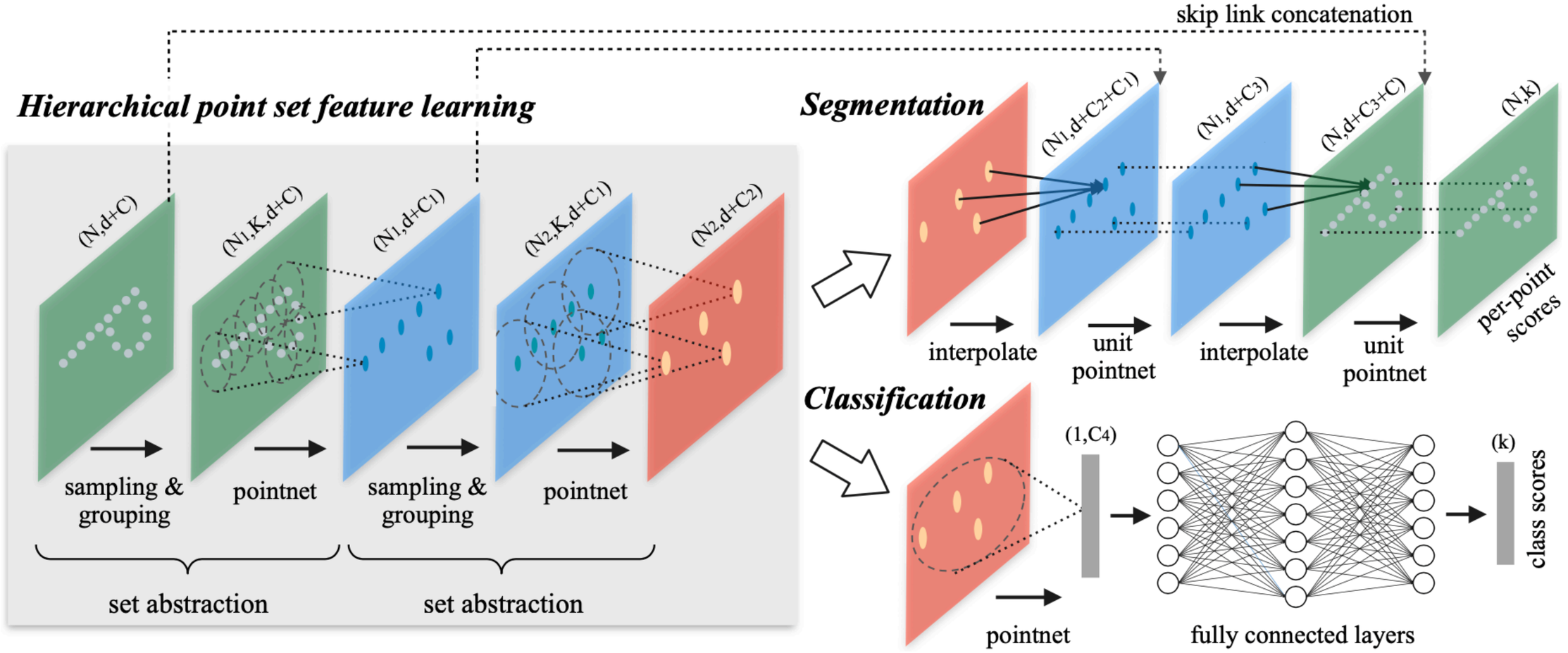


Concatenating point representations and global representation is not enough to capture local structures

PointNet++

PointNet ++

- PointNet++ incorporates **hierarchical** structures and applies **PointNet recursively** to capture **local and global features** effectively.

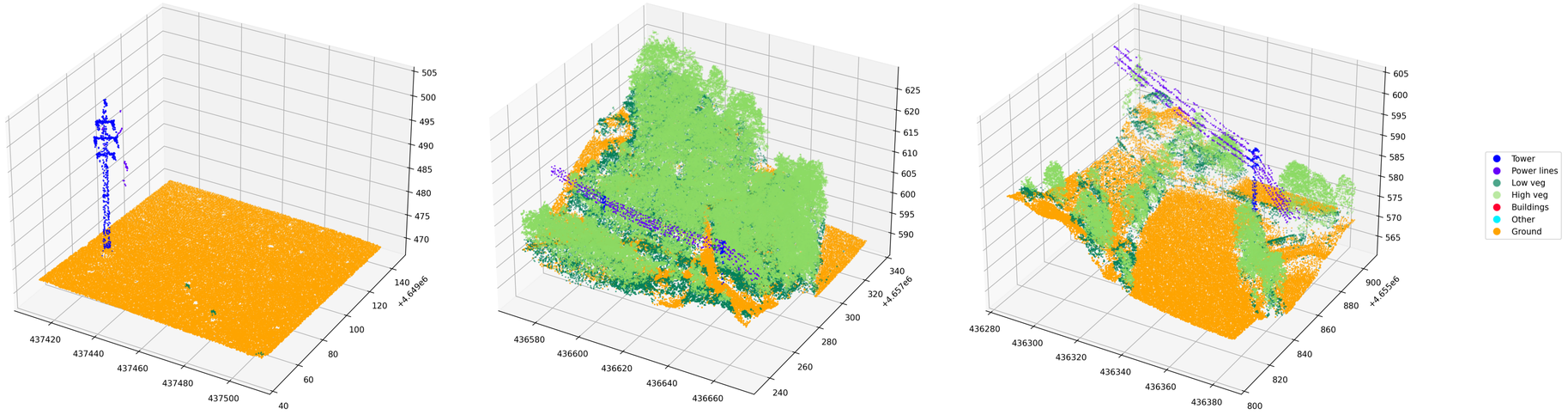


More strategies for adding inductive biases

- **Color dropout:** Randomly removing color information
  - It forces the network to focus more on the geometric relationships between points
- **Height Appending**
  - Adding height information to each point can be valuable, especially in tasks where vertical information is crucial, such as recognizing objects with varying heights or understanding the overall structure of a scene.

Open problems

- Network should be invariant to density
- Adding color uncertainty depending on the point's location
- Overconfident. Shows little doubt when making incorrect predictions.
- Not designed for imbalanced data.



Ressources

- Medium DataScienceUB: Pointnet implementation explained
- PointNet notebook
- PointNet paper
- PointNet++ paper
- PointNeXt paper