3D representations in machine learning

GEO1004:
3D modelling of the built environment

https://3d.bk.tudelft.nl/courses/geo1004

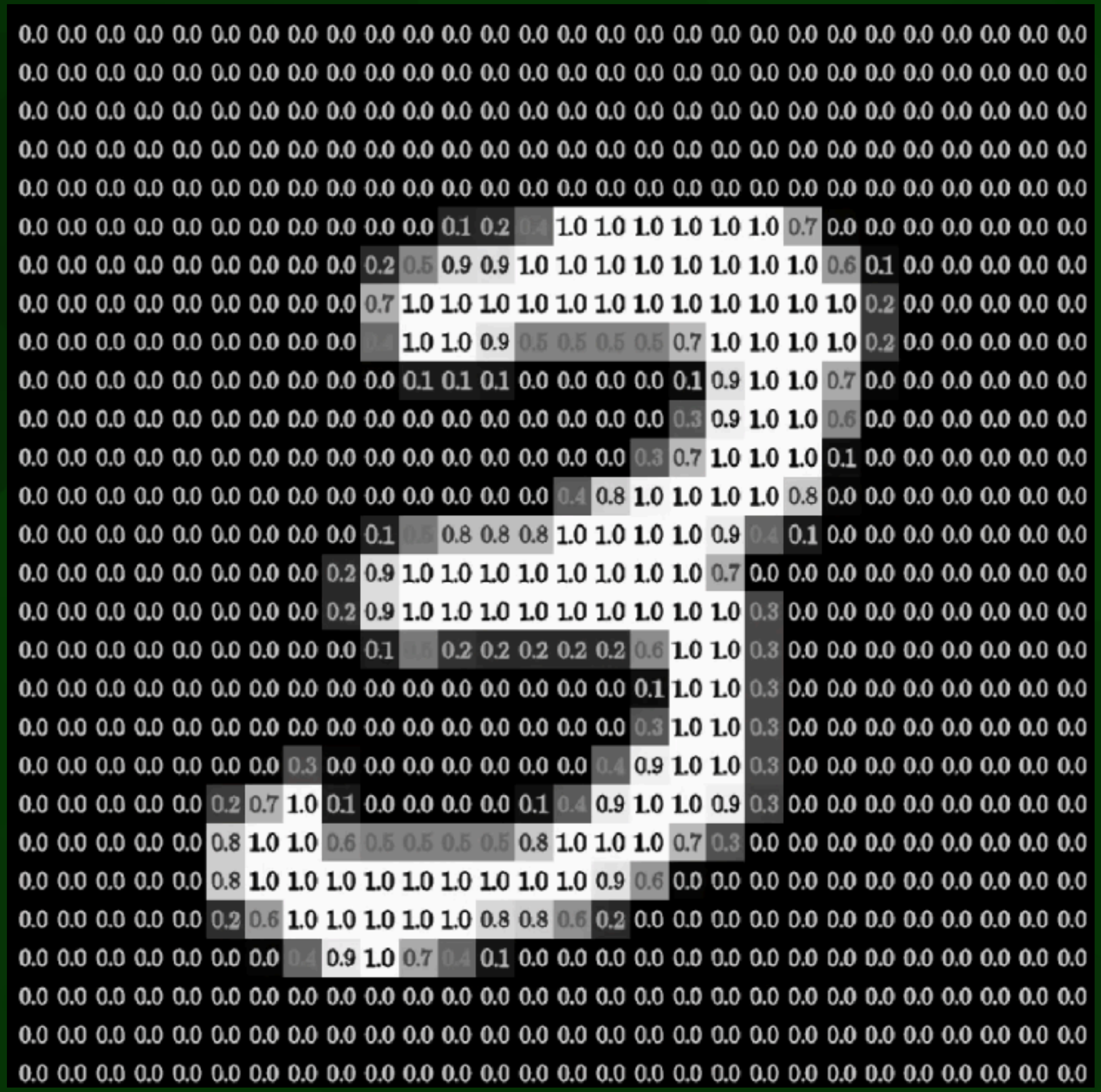
3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

What is ML / DL?

- Instead of explicitly programming rules to solve a problem...

- use a system that attempts to learn these rules automatically using a large data set where the problem has been solved (training data).

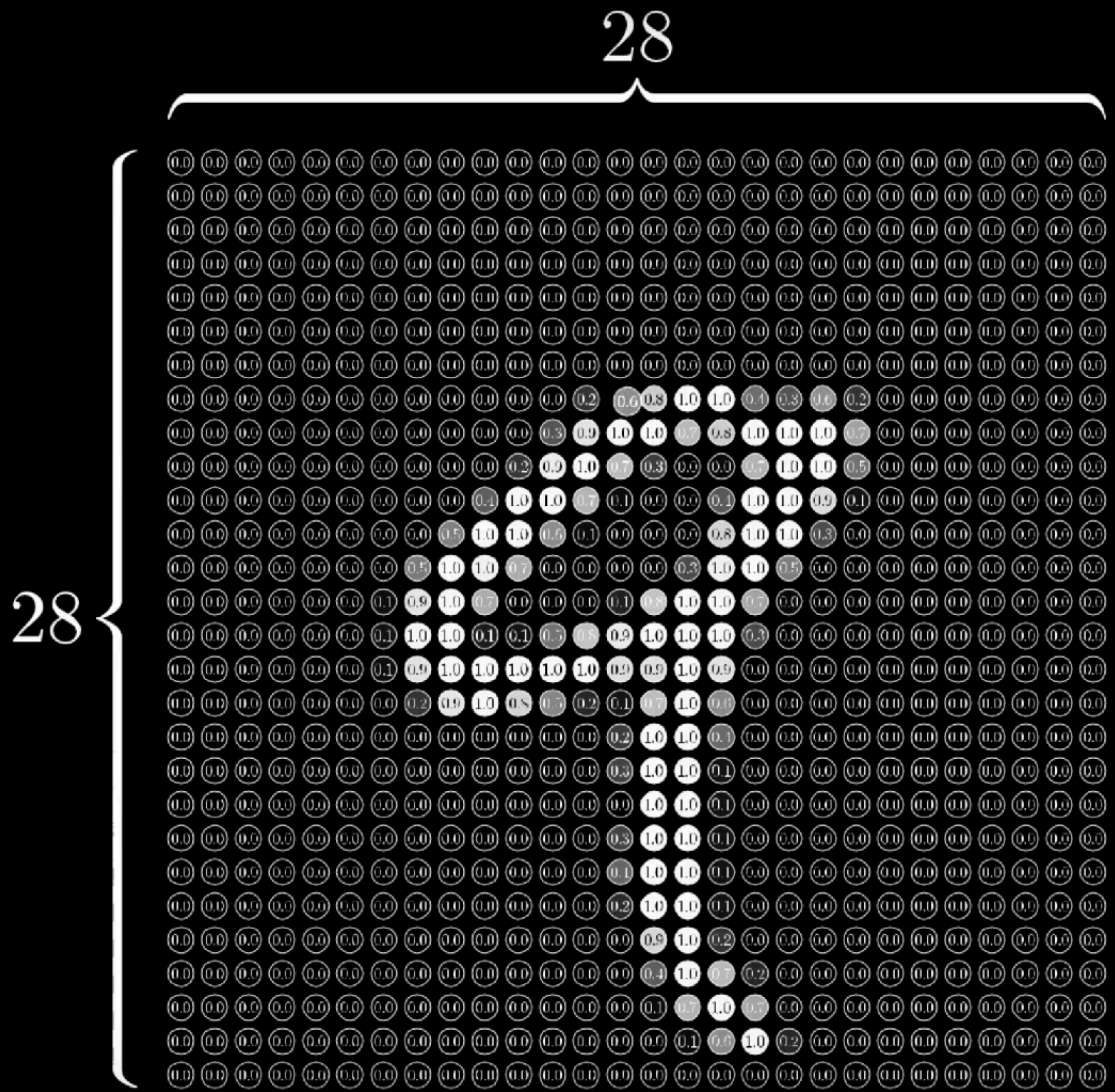
- Many such systems: regression, support vector machines (SVMs), decision trees, neural networks, clustering, principal component analysis (PCA), etc.



3blue1brown - Youtube

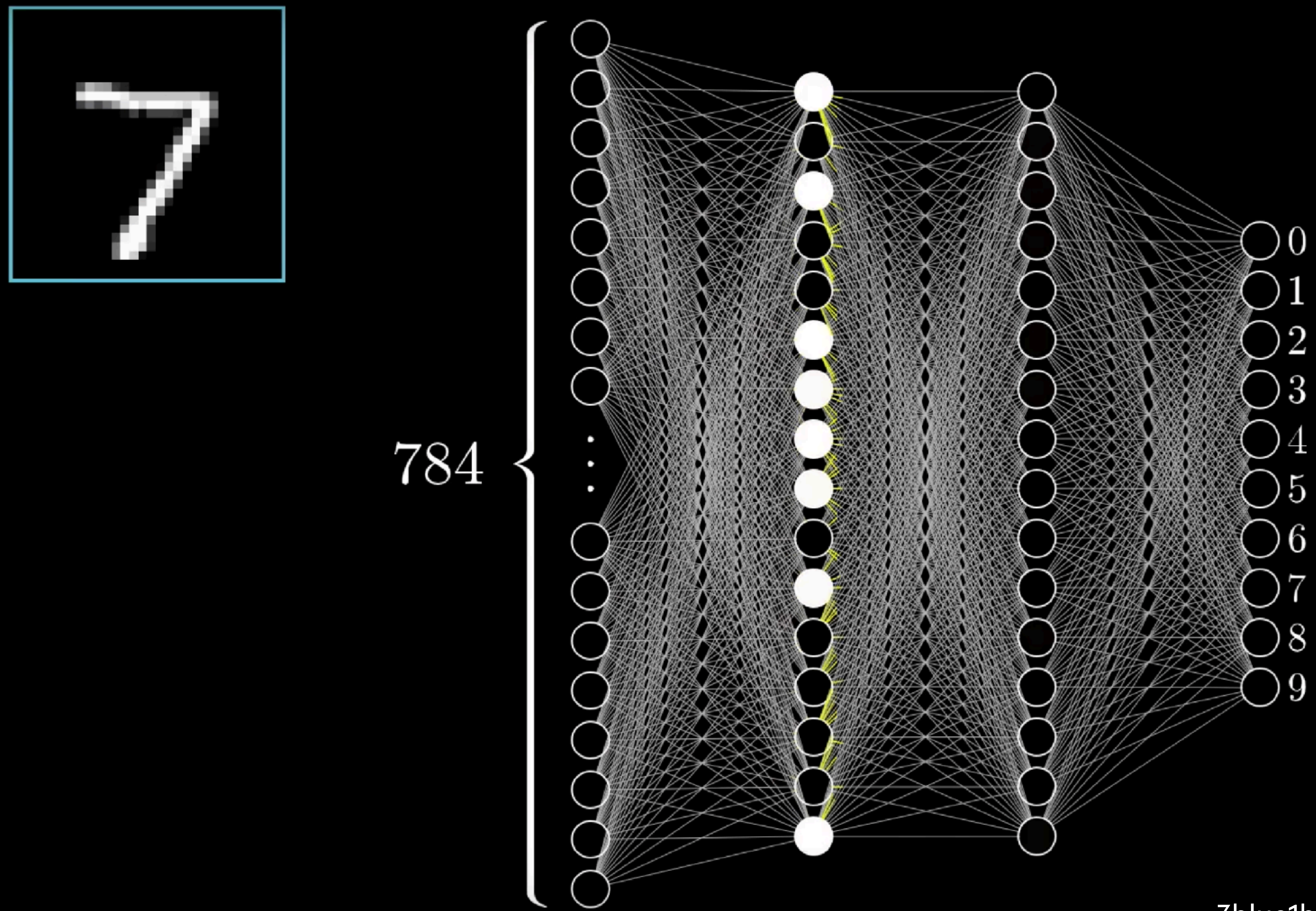
What is ML / DL?

- Traditional machine learning: manually design features that meaningfully represent the characteristics of the data, then pass on the features for each data point to the system
- Deep learning: pass the data as-is to a more complex system (deep neural network) which attempts to learn the features of the data on its own



$$28 \times 28 = 784$$

3blue1brown - Youtube



3blue1brown - Youtube

Successes of ML / DL

- Face and object recognition, e.g. finding pictures of certain people in your phone
- Speech recognition and natural language, e.g. AI assistants
- Translation between languages
- Playing certain games, e.g. classical board games
- Fraud detection, e.g. detecting suspicious card payments
- (Hand)writing recognition, e.g. digitising books

ML / DL with 2D geodata

- Extracting features from imagery, e.g. buildings, roads, vegetation, etc.
- Pattern recognition, e.g. image classification, clustering features, etc.
- Weather / climate predictions, e.g. temperature or rain for next day
- Routing, e.g. finding optimal routes for different transport modes

What about 3D geodata?

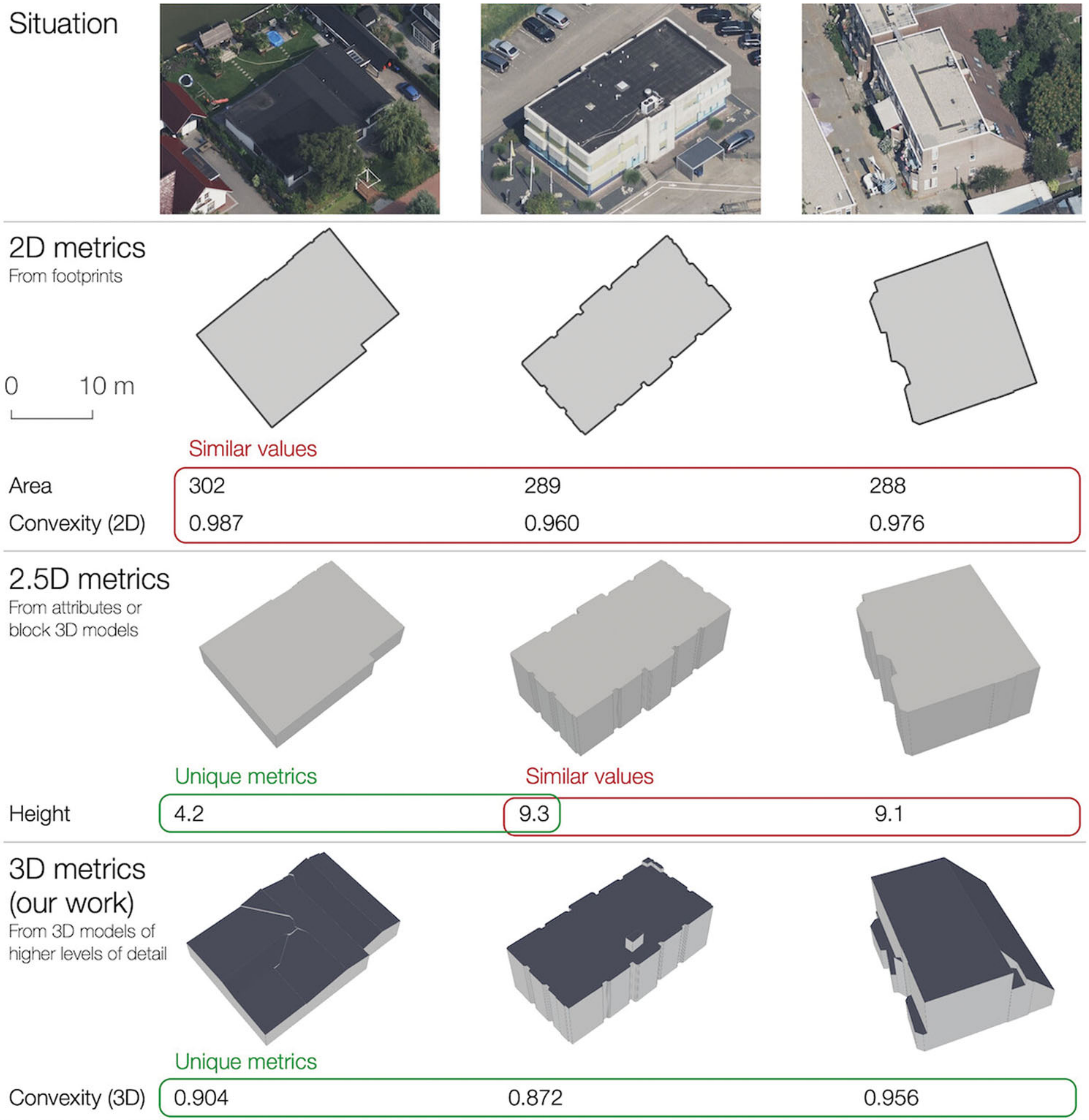
What makes 3D difficult?

- Voxels: very large sizes for good resolution -> methods for images don’t work as well
  - Trees: complex structure -> usually not suitable for ML / DL
- Point clouds: very large sizes, no explicit structure
- Meshes: easy to modify vertices but topology not so much -> problems with geometric errors or watertightness

Today's lecture

- Three very different approaches to deal with 3D data:
  - Machine learning with 3D features
  - Neural networks for point clouds
  - Implicit field representations

3D building metrics for urban morphology



Situation
2D metrics
From footprints
0
10 m
Similar values
Area
302
289
288
Convexity (2D)
0.987
0.960
0.976
2.5D metrics
From attributes or block 3D models
Unique metrics
Similar values
Height
4.2
9.3
9.1
3D metrics (our work)
From 3D models of higher levels of detail
Unique metrics
Convexity (3D)
0.904
0.872
0.956

Table 1. Metrics are computed per building based on category.

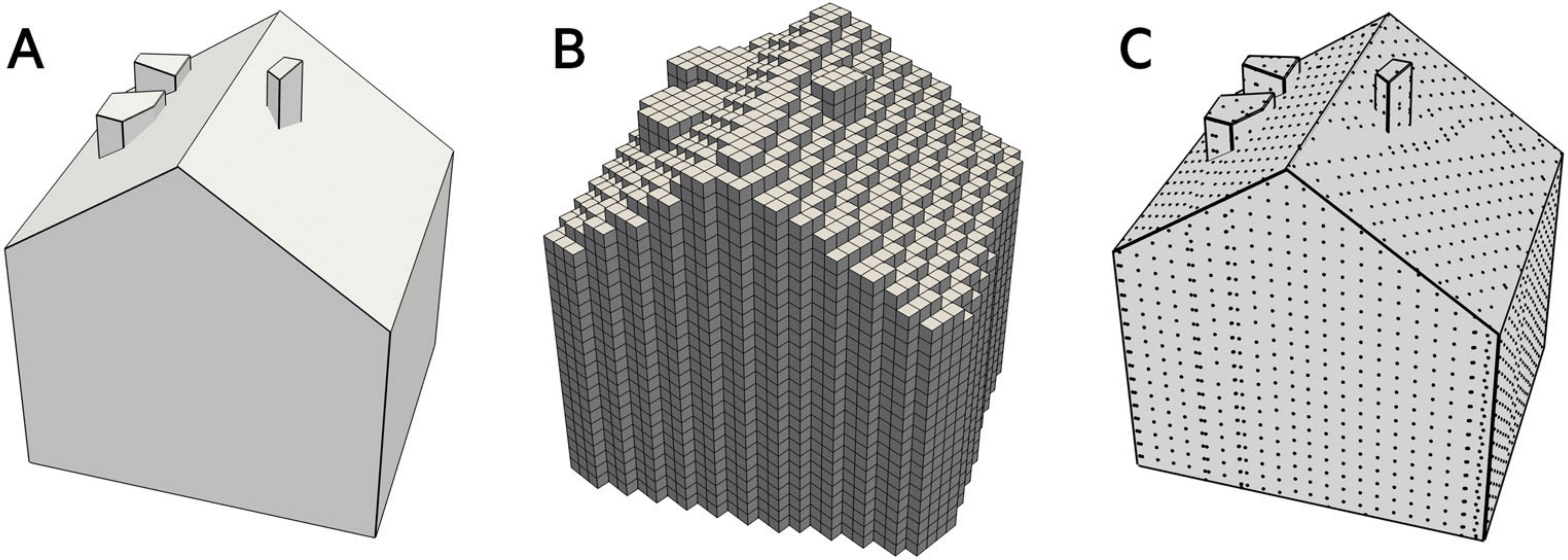
Geometric properties	Number of vertices, Number of surfaces, Number of vertices by semantic type (i.e. ground, roof, wall), Number of surfaces by semantic type (i.e. ground, roof, wall), Min/Max/Range/Mean/Median/Std/Mode height
Derived properties	Footprint perimeter, Volume, Volume of convex hull, Volume of Object-Oriented Bounding Box, Volume of Axis-Oriented Bounding Box, Volume of voxelised building, Length and width of the Object-Oriented Bounding Box, Surface area, Surface area by semantic surface, Horizontal elongation, Min/Max vertical elongation, Form factor
Spatial distribution	Shared walls, Nearest neighbour
Space indices (see Table 3)	Circularity/Hemisphericality*, Convexity 2D/3D*, Fractality 2D/3D*, Rectangularity/Cuboidness*, Squareness/Cubeness*, Cohesion 2D/3D*, Proximity 2D/3D^{+}, Exchange 2D/3D^{+}, Spin 2D/3D^{+}, Perimeter/Circumference*, Depth 2D/3D^{+}, Girth 2D/3D^{+}, Dispersion 2D/3D^{x}, Range 2D/3D*, Equivalent Rectangular/Cuboid*, Roughness^{x}

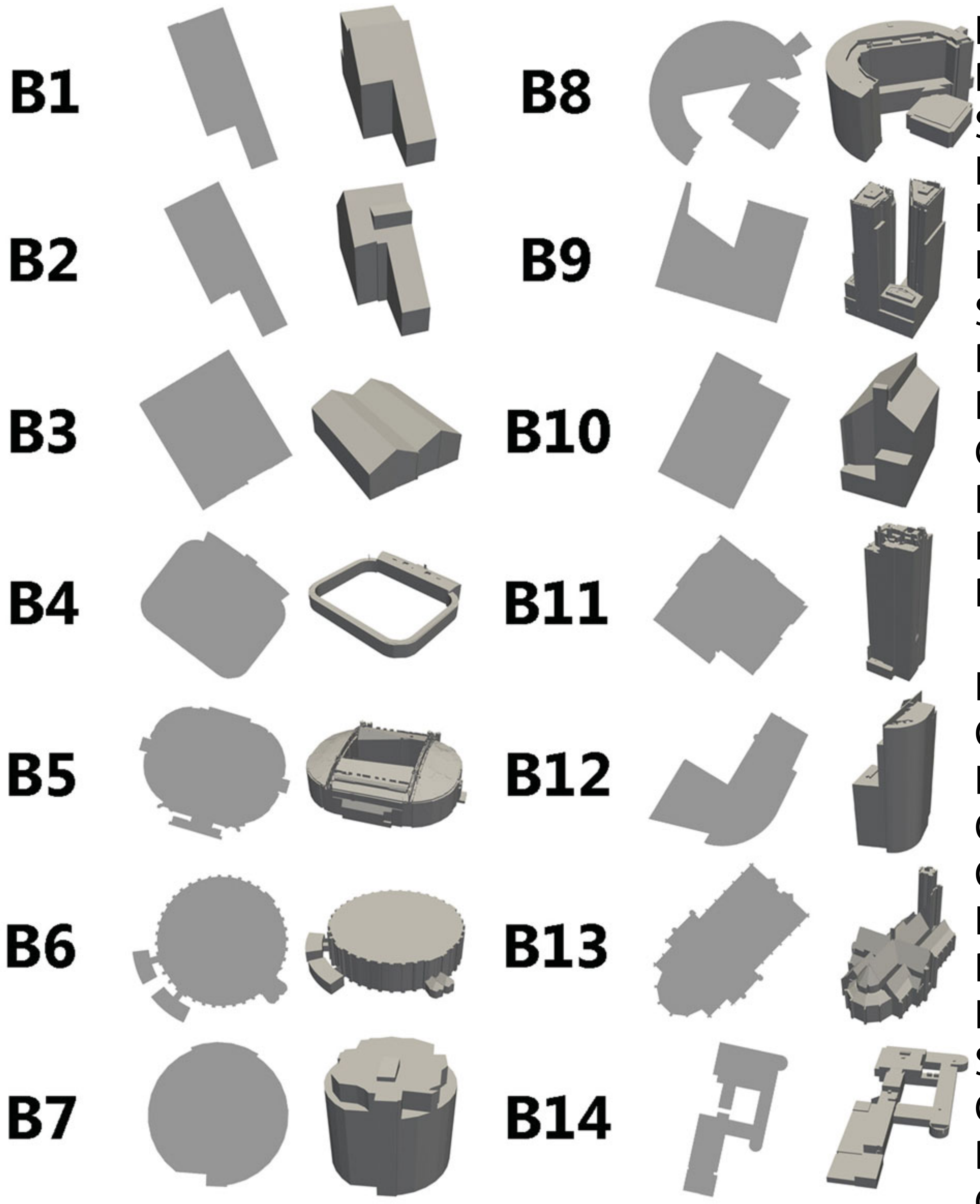
*Formula-based index, size-independent by definition.

$^{+}$Index based on interior grid points (discretised), normalised.

^{x}Index based on surface grid points (discretised), normalised.

	2D	3D
Circularity/ Hemisphericality	It measures the area deviation between a polygon and its equal-perimeter circle. Circle is generally assumed as the most compact shape.	It measures the volume deviation between a polyhedron and its equal-area hemisphere. A hemisphere was selected to represent the space above ground.
Convexity	It measures the area deviation between a polygon and its convex hull. Thus, it reveals a polygon's degree of being curved inward or outward.	It measures the volume deviation between a polyhedron and its convex hull. Thus, it reveals a building's degree of being curved inward or outward.
Fractality	It measures the edge roughness or smoothness. Based on Wentz (2010).	It measures the surface roughness or smoothness.
Rectangularity/ Cuboidness	It measures the area deviation between a polygon and its minimum area bounding rectangle. Thus, it reveals a polygon's degree of being curved inward.	It measures the volume deviation between a polyhedron and its minimum volume bounding box. Thus, it reveals a polyhedral's degree of being curved inwards.
Squareness/ Cubeness	It measures the perimeter deviation between a polygon and its equal-area square.	It measures the surface area deviation between a polyhedron and its equal-volume cube.
Cohesion	It is a measure of overall accessibility from all points to others within a polygon.	It is a measure of overall accessibility from all points to others within a polyhedron.
Proximity	It is a measure of overall accessibility from all inner points to the centre of a polygon.	It is a measure of overall accessibility from all inner points to the centre of a polyhedron.


A
B
C



	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14
							2D							
Circularity	0.52	0.51	0.76	0.83	0.48	0.28	0.91	0.29	0.46	0.72	0.64	0.51	0.39	0.20
Convexity	0.92	0.90	0.99	0.98	0.94	0.86	0.98	0.73	0.84	0.98	0.94	0.80	0.86	0.68
Fractality	0.43	0.45	0.36	0.21	0.24	0.32	0.32	0.32	0.28	0.40	0.31	0.30	0.30	0.32
Rectangularity	0.85	0.81	0.99	0.92	0.74	0.69	0.79	0.58	0.83	0.96	0.89	0.61	0.72	0.58
Squareness	0.81	0.81	0.98	1.03	0.78	0.60	1.08	0.61	0.76	0.96	0.90	0.80	0.70	0.50
Equivalent rectangularity	0.92	0.90	0.99	1.03	0.78	0.60	1.08	0.61	0.77	0.98	0.90	0.82	0.73	0.54
Proximity	0.82	0.83	0.98	0.99	0.99	0.98	1.00	0.84	0.91	0.95	0.98	0.85	0.92	0.67
Exchange	0.68	0.68	0.90	0.92	0.93	0.90	0.98	0.75	0.81	0.83	0.92	0.74	0.78	0.53
Spin	0.61	0.61	0.94	0.97	0.97	0.94	1.00	0.71	0.81	0.88	0.95	0.69	0.80	0.43
Perimeter	0.72	0.71	0.87	0.91	0.69	0.53	0.96	0.54	0.68	0.85	0.80	0.71	0.62	0.45
Depth	0.67	0.67	0.88	0.91	0.88	0.84	0.97	0.52	0.67	0.83	0.85	0.64	0.74	0.41
Girth	0.56	0.60	0.82	0.84	0.83	0.92	0.90	0.51	0.58	0.70	0.79	0.52	0.67	0.44
Dispersion	0.70	0.70	0.90	0.93	0.93	0.91	0.98	0.75	0.79	0.84	0.91	0.74	0.79	0.56
Range	0.59	0.59	0.79	0.87	0.87	0.85	0.97	0.73	0.73	0.75	0.78	0.63	0.71	0.54
Roughness	0.70	0.69	0.82	0.85	0.49	0.29	0.93	0.29	0.50	0.80	0.67	0.57	0.43	0.21
							3D							
Hemisphericality	0.42	0.43	0.41	0.15	0.17	0.29	0.52	0.28	0.20	0.44	0.21	0.30	0.25	0.14
Convexity	0.88	0.89	0.91	0.23	0.67	0.85	0.94	0.58	0.59	0.87	0.85	0.74	0.55	0.49
Fractality	0.32	0.33	0.31	0.25	0.21	0.23	0.24	0.23	0.22	0.30	0.24	0.21	0.23	0.27
Cuboidness	0.62	0.60	0.83	0.17	0.43	0.64	0.70	0.42	0.52	0.70	0.73	0.48	0.27	0.30
Cubeness	0.88	0.90	0.87	0.44	0.49	0.68	1.00	0.66	0.54	0.90	0.56	0.71	0.62	0.42
Equivalent cuboidness	0.96	0.97	0.96	0.61	0.57	0.90	1.01	0.69	0.57	0.93	0.64	0.76	0.64	0.66
Proximity	0.83	0.83	0.78	0.32	0.68	0.76	0.96	0.76	0.73	0.89	0.69	0.75	0.78	0.41
Exchange	0.71	–	0.57	0	–	0.48	0.89	–	–	0.78	–	0.52	–	0.14
Spin	0.65	0.66	0.58	0.10	0.46	0.55	0.91	0.56	0.51	0.78	0.43	0.51	0.56	0.15
Circumference	0.71	0.72	0.70	0.36	0.39	0.55	0.81	0.53	0.44	0.73	0.45	0.57	0.5	0.34
Depth	0.59	0.59	0.55	0.32	0.40	0.55	0.79	0.5	0.43	0.63	0.32	0.51	0.49	0.30
Girth	0.55	0.55	0.43	0.40	0.37	0.34	0.80	0.47	0.44	0.64	0.30	0.36	0.44	0.30
Dispersion	0.79	0.79	0.74	0.40	0.73	0.71	0.93	0.75	0.69	0.85	0.63	0.69	0.71	0.47
Range	0.57	0.60	0.60	0.32	0.57	0.58	0.80	0.59	0.54	0.65	0.47	0.50	0.51	0.28
Roughness	0.87	0.87	0.91	3.54	0.43	0.84	0.79	0.63	0.47	0.77	0.58	0.79	0.65	1.03
Form factor	1.11	1.04	2.04	8.87	2.75	2.55	1.04	1.55	0.83	0.98	0.49	0.58	1.84	4.65

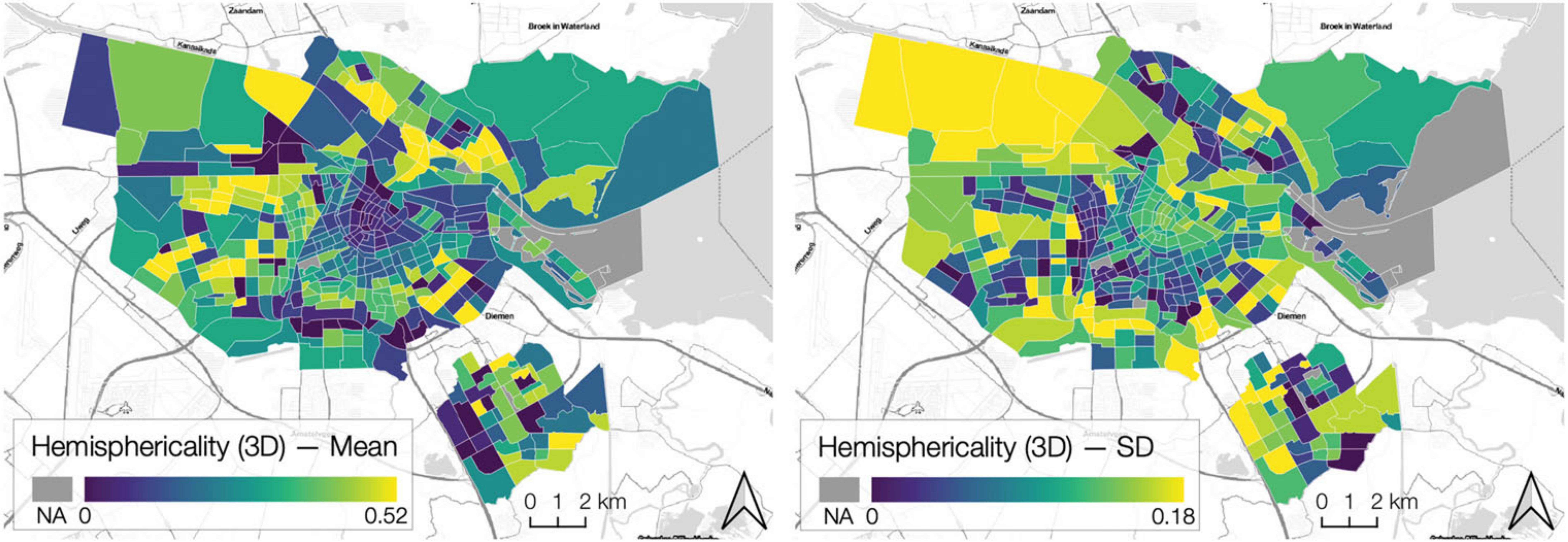
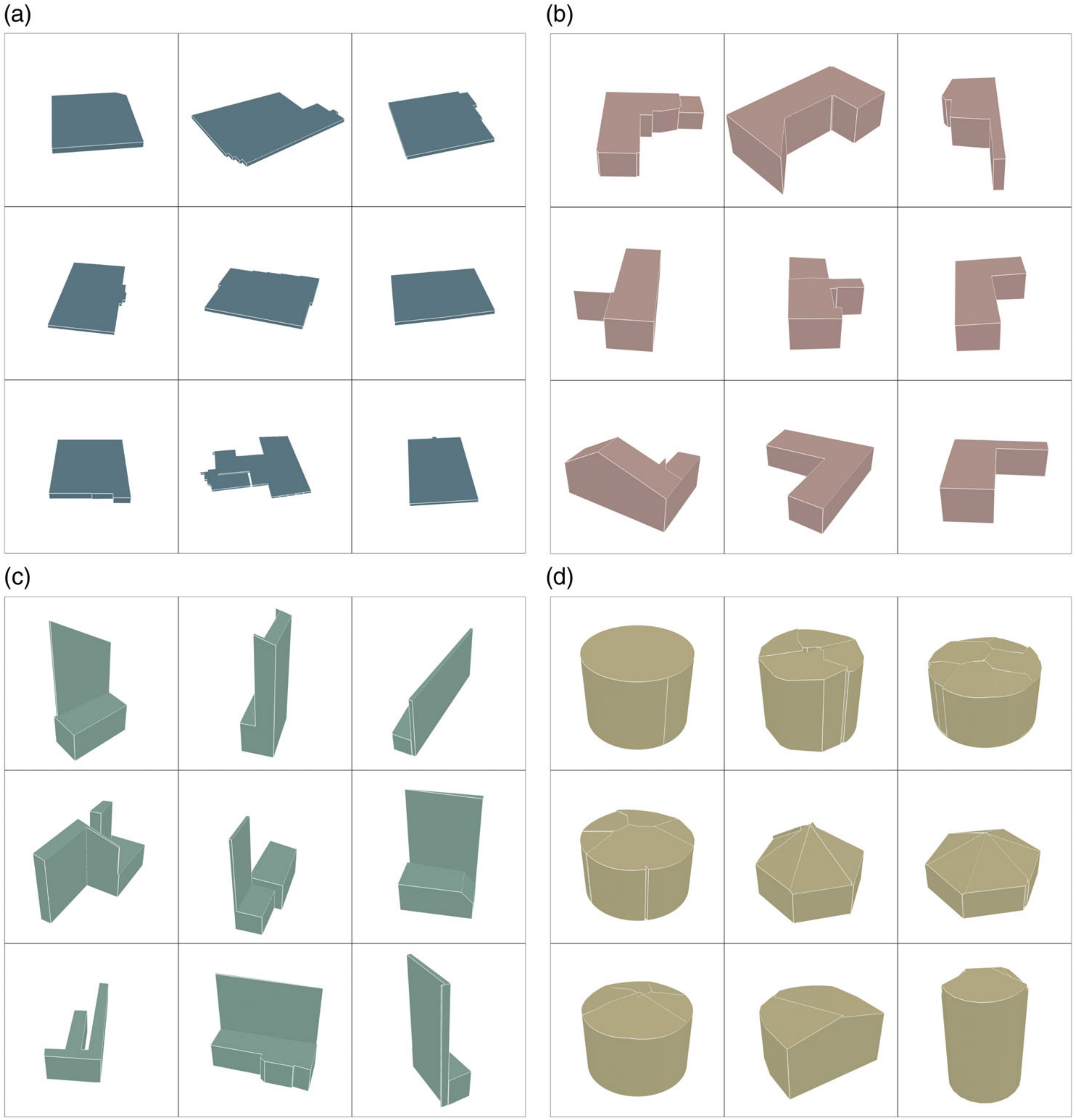
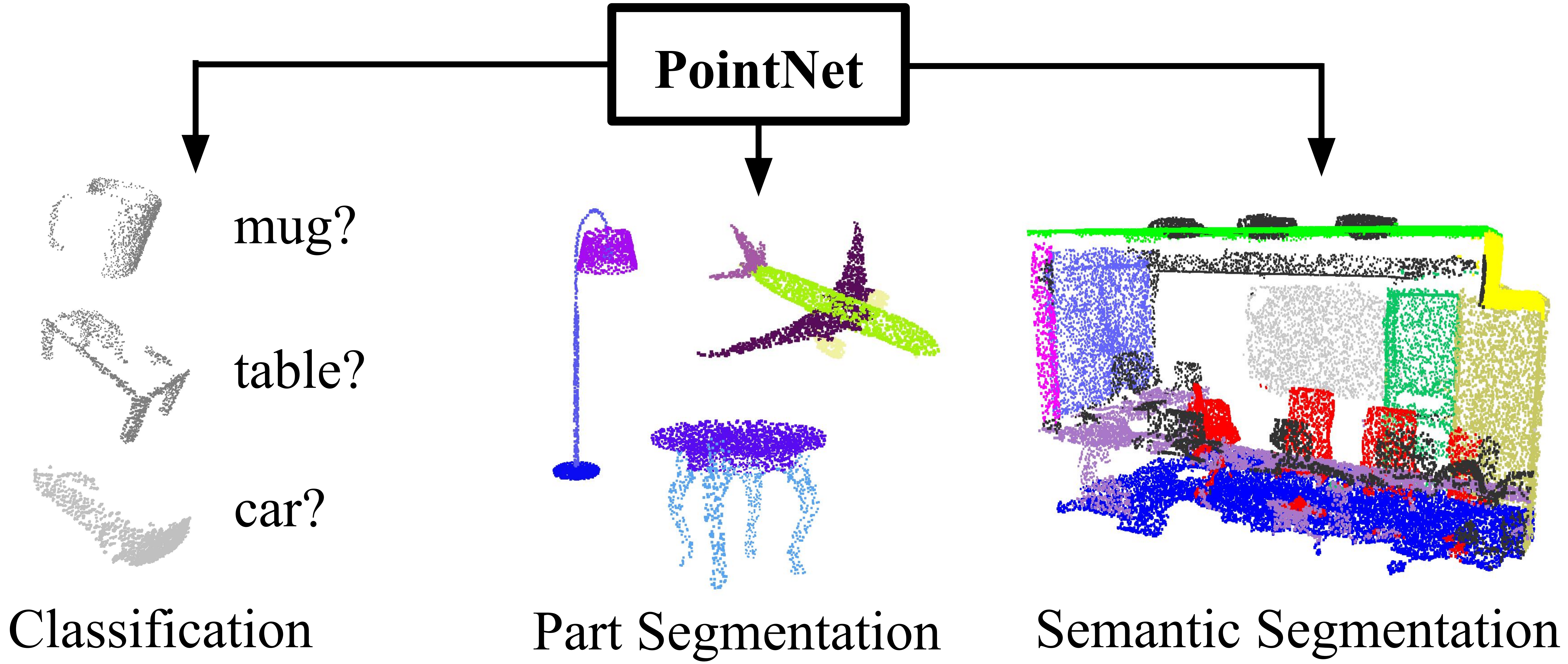


Figure 5. The 3D urban form of a city. Aggregations of one of the 3D metrics (*hemisphericality*) at the level of the administrative neighbourhoods (*buurt*) of the municipality of Amsterdam—mean as a measure of central tendency (left); and another one (standard deviation) indicating its dispersion within each area (right). The maps highlight the difference between the historical city centre with the well-known canal houses (ranking low on the *hemisphericality* index) and the newer neighbourhoods. The administrative boundaries and the basemap are courtesy of Statistics Netherlands, Stamen, and OpenStreetMap contributors.

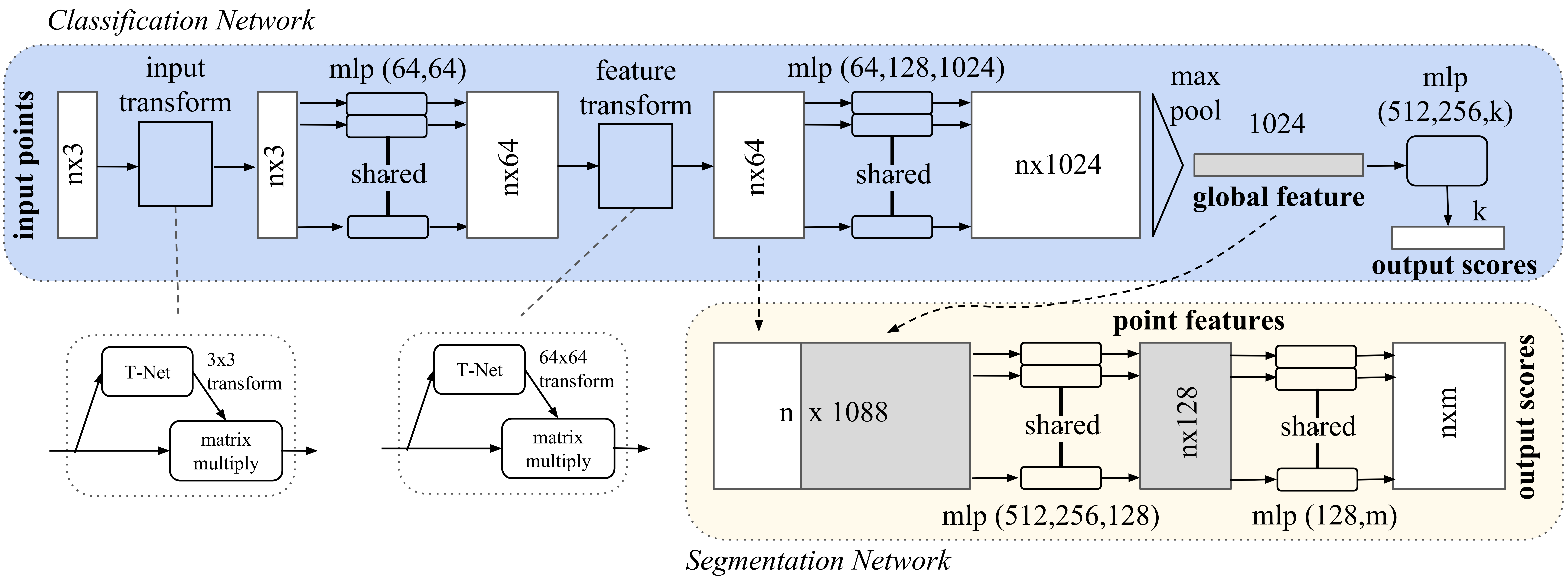

(a)
(b)
(c)
(d)

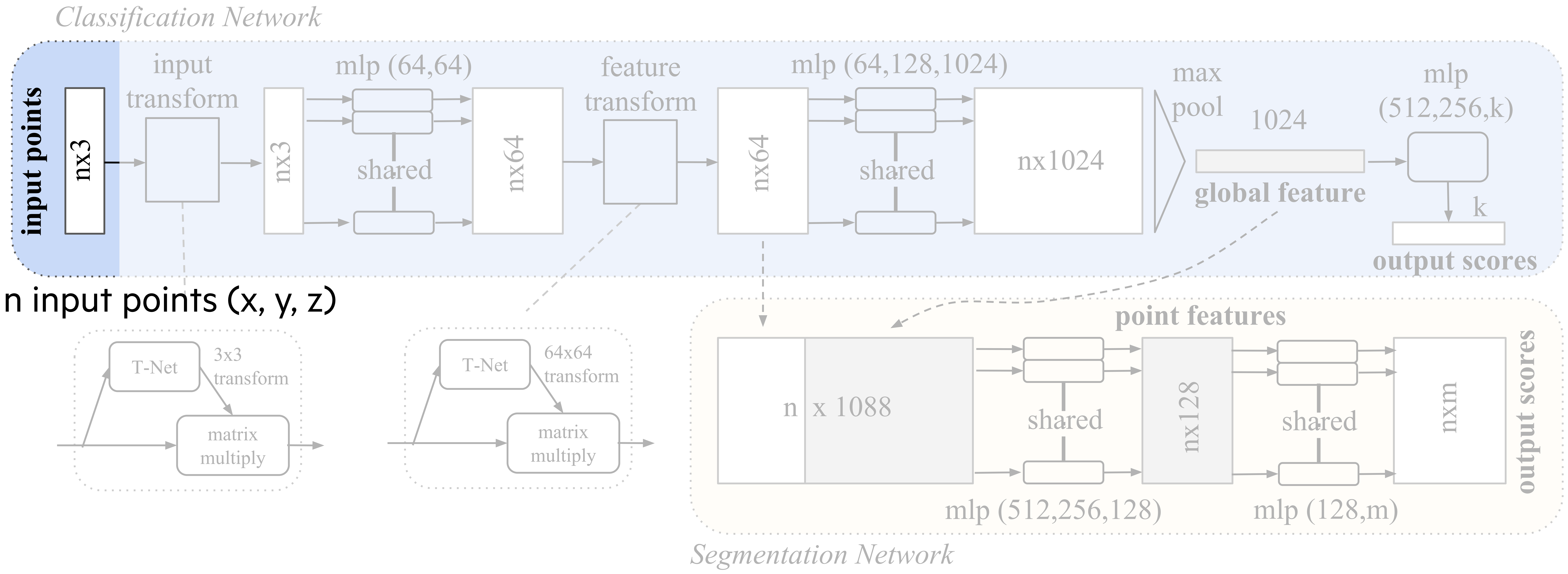
PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

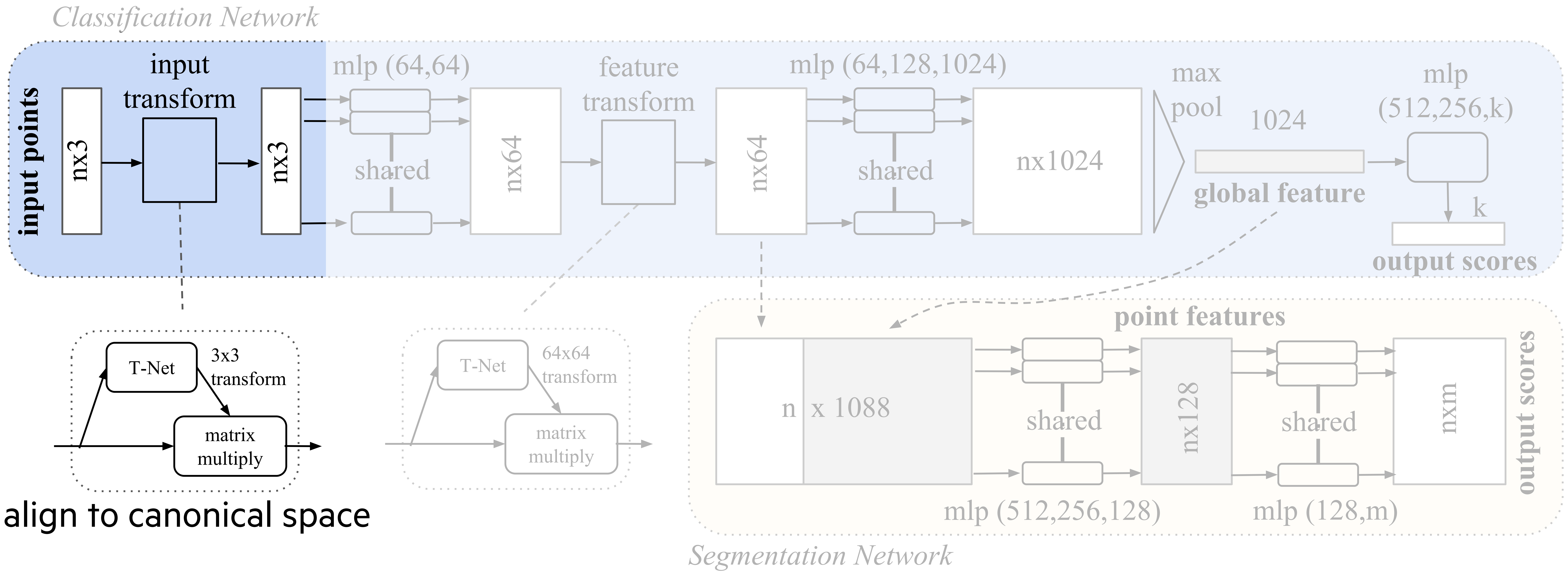

PointNet
mug?
table?
car?
Classification
Part Segmentation
Semantic Segmentation

Point clouds vs. neural networks

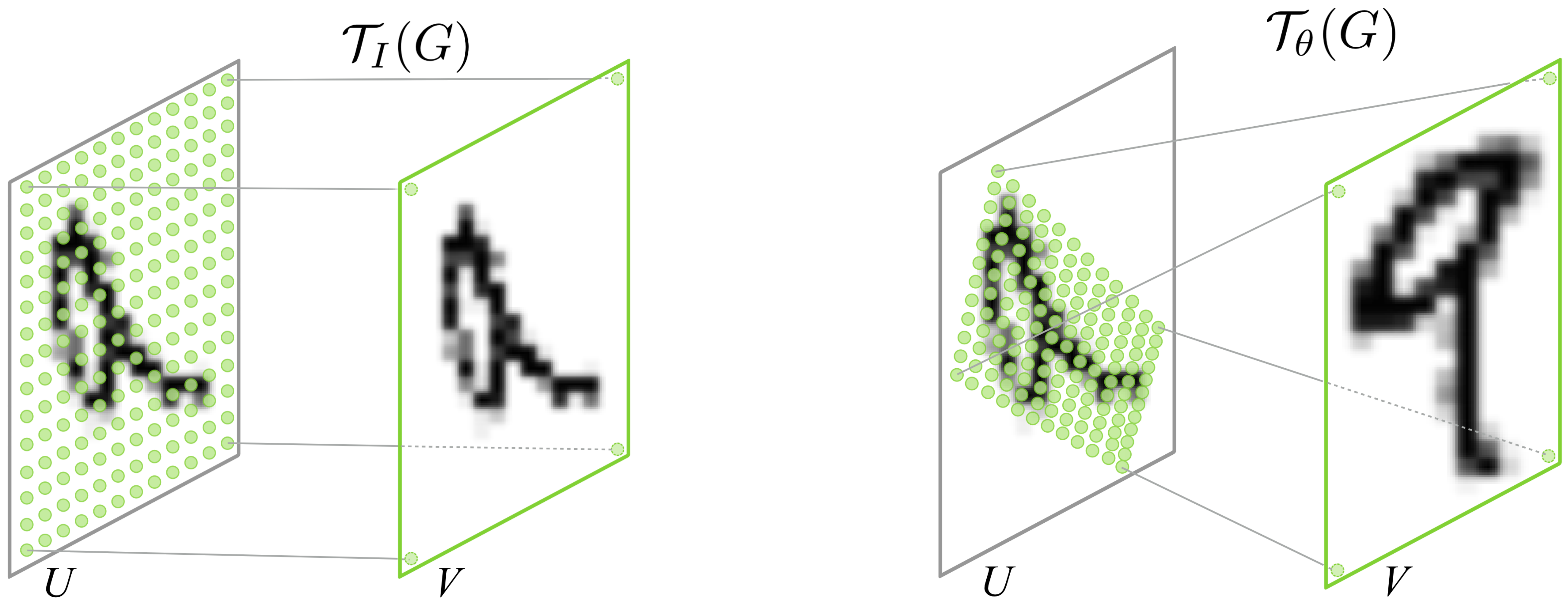
- Unordered: reordering points in a point cloud shouldn't affect the result
- Transformation invariance: rotating or translating a point cloud shouldn't affect the result
- Local structure: points need to be analysed together with other nearby points (not in isolation)


Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

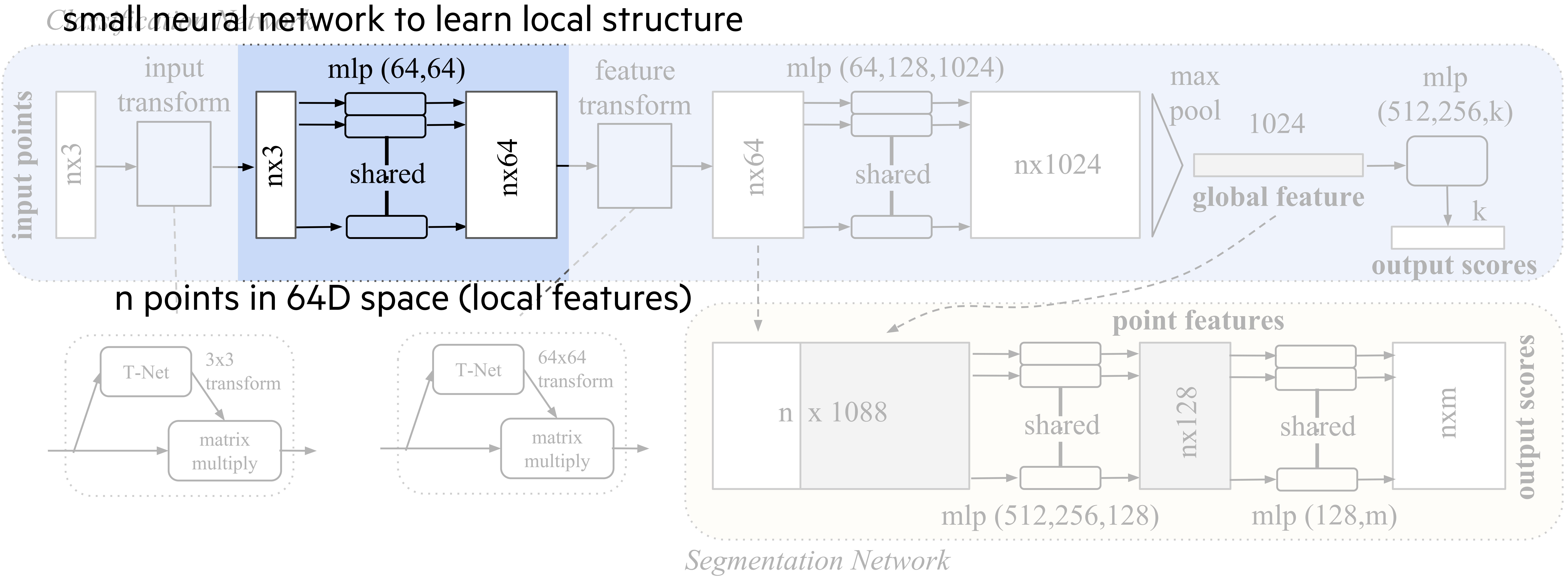

Classification Network
input points
nx3
input transform
mlp (64,64)
nx3
shared
nx64
feature transform
mlp (64,128,1024)
nx64
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
n input points (x, y, z)
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

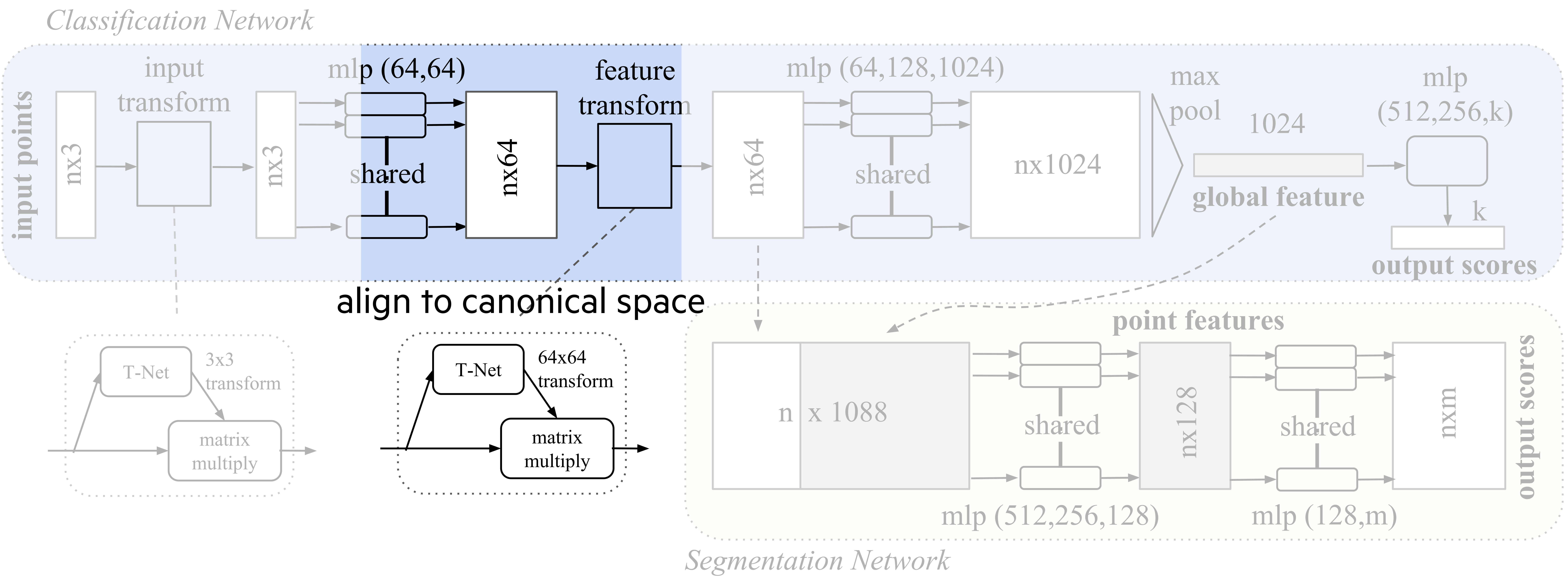


Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
T-Net
3x3 transform
matrix multiply
align to canonical space
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

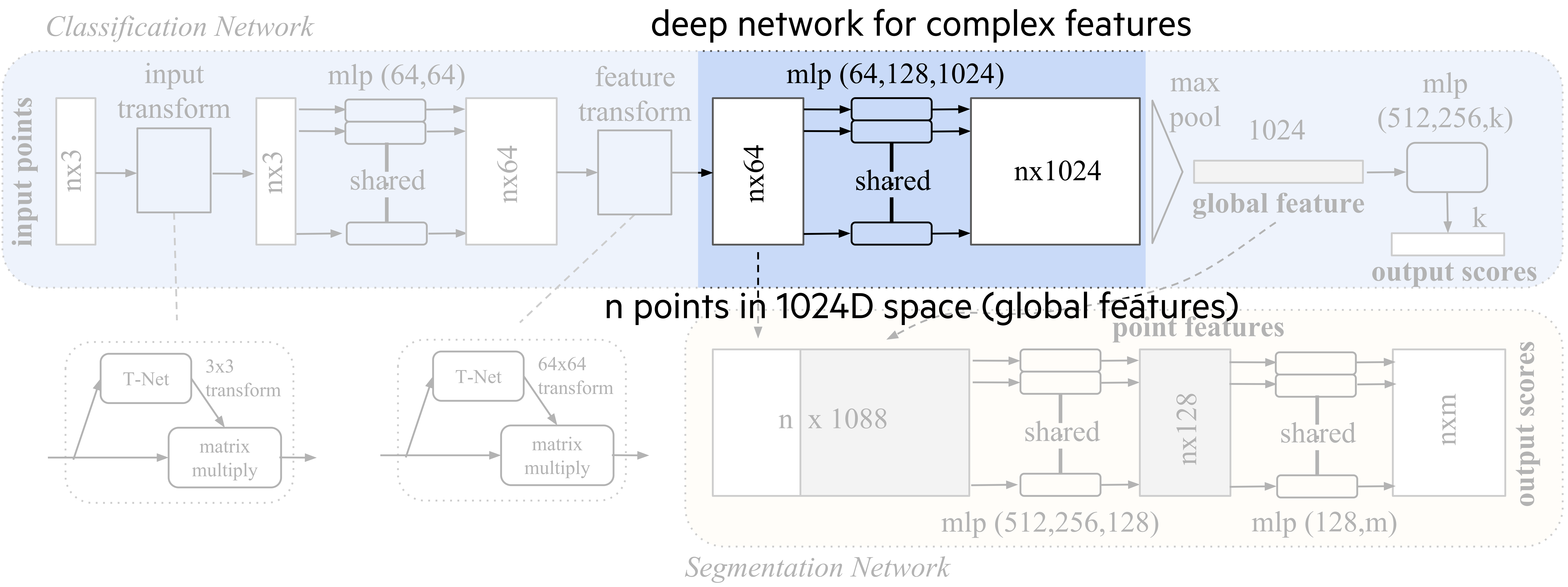


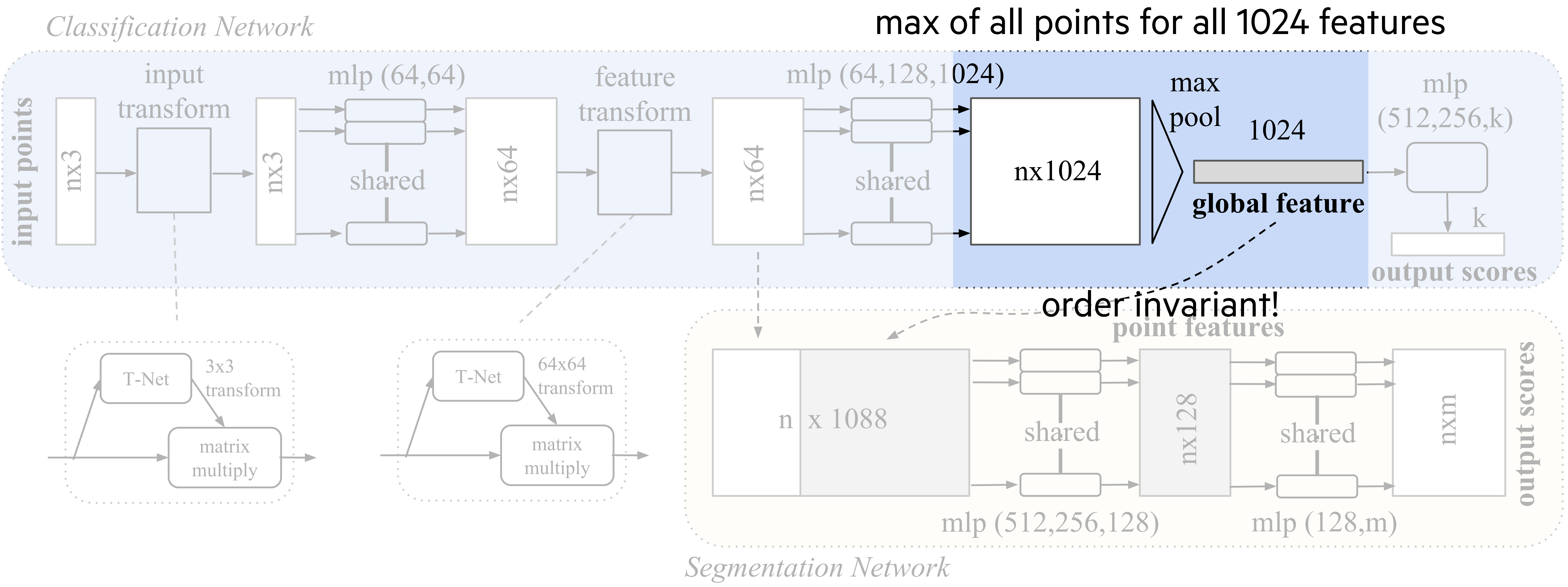
(a) (b)


small neural network to learn local structure
n points in 64D space (local features)
input points
nx3
input transform
mlp (64,64)
nx3
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network



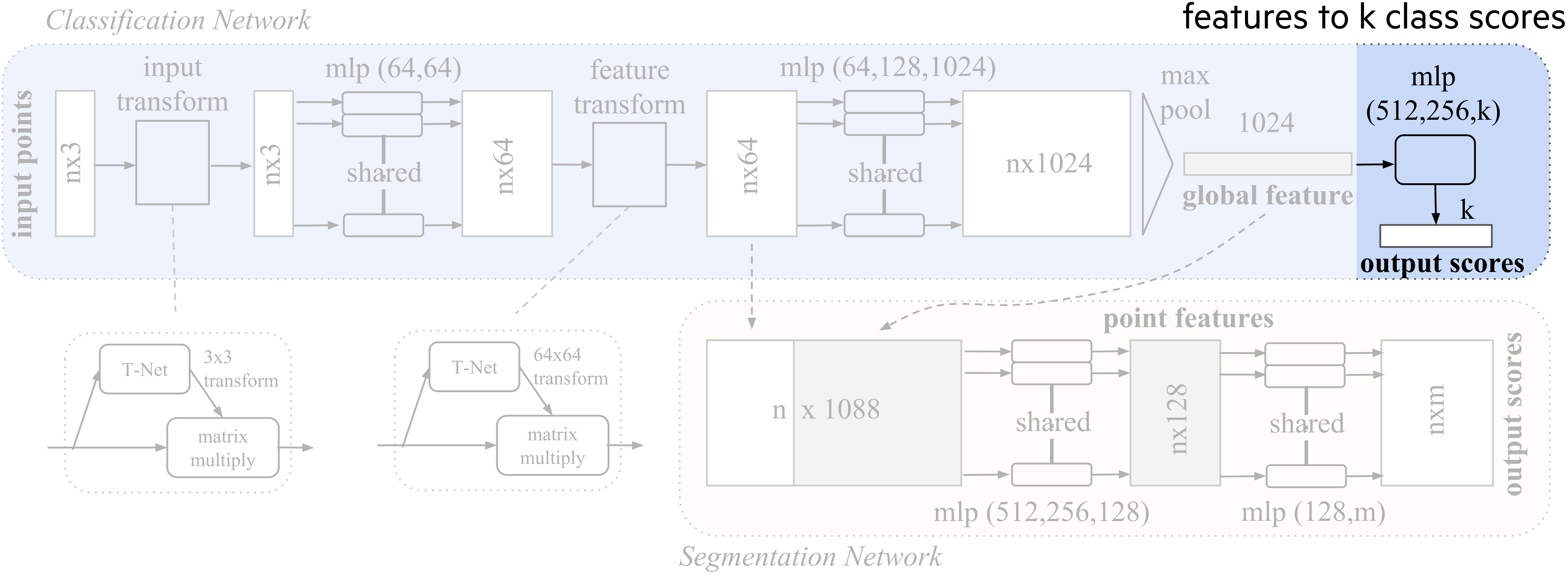
Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
align to canonical space
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

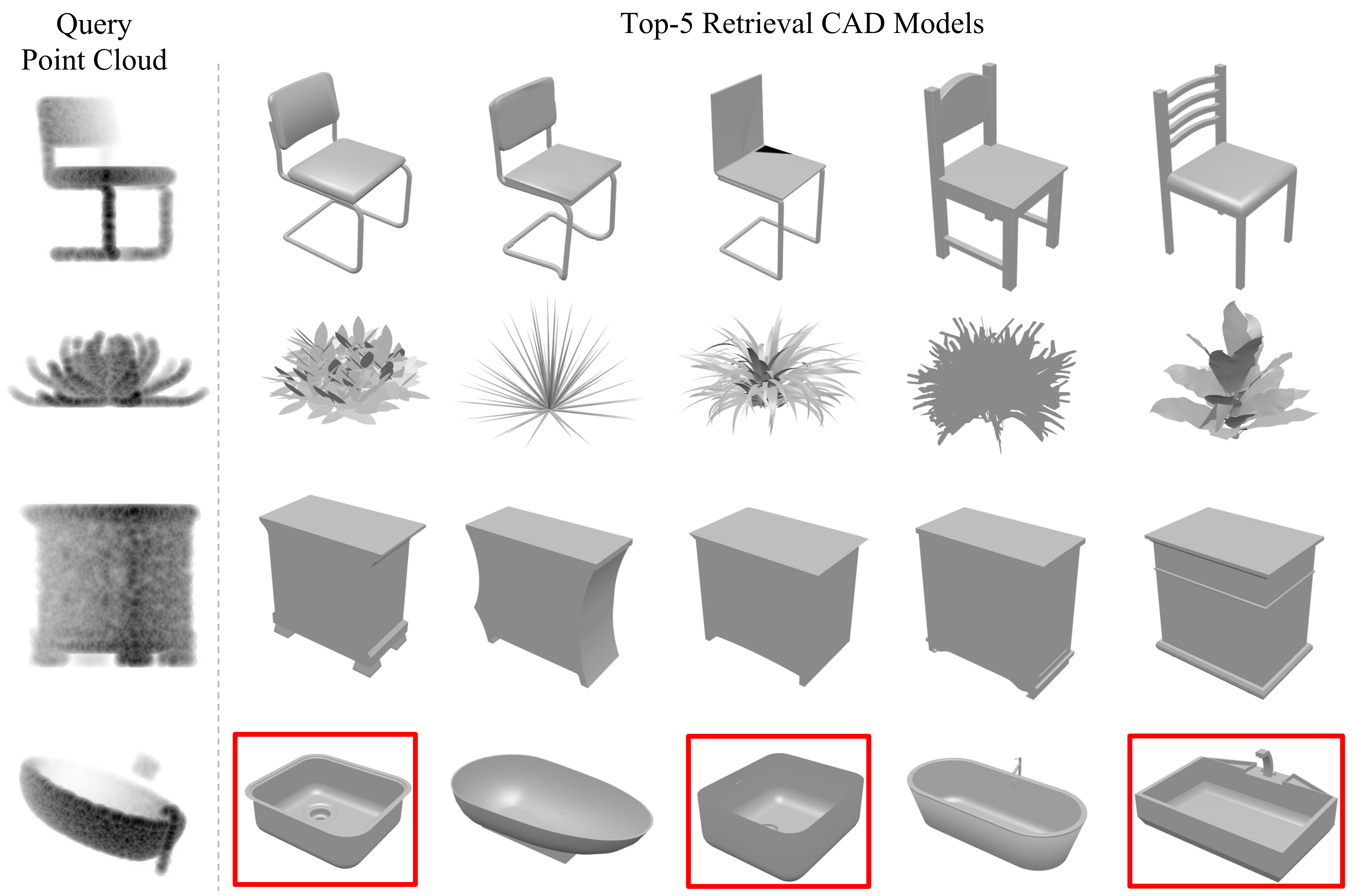

deep network for complex features
Classification Network
input points
nx3
input
transform
nx3
mlp (64,64)
shared
nx64
feature
transform
mlp (64,128,1024)
nx64
shared
nx1024
max
pool
1024
global feature
mlp
(512,256,k)
k
output scores
n points in 1024D space (global features)
point features
T-Net
3x3
transform
matrix
multiply
T-Net
64x64
transform
matrix
multiply
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

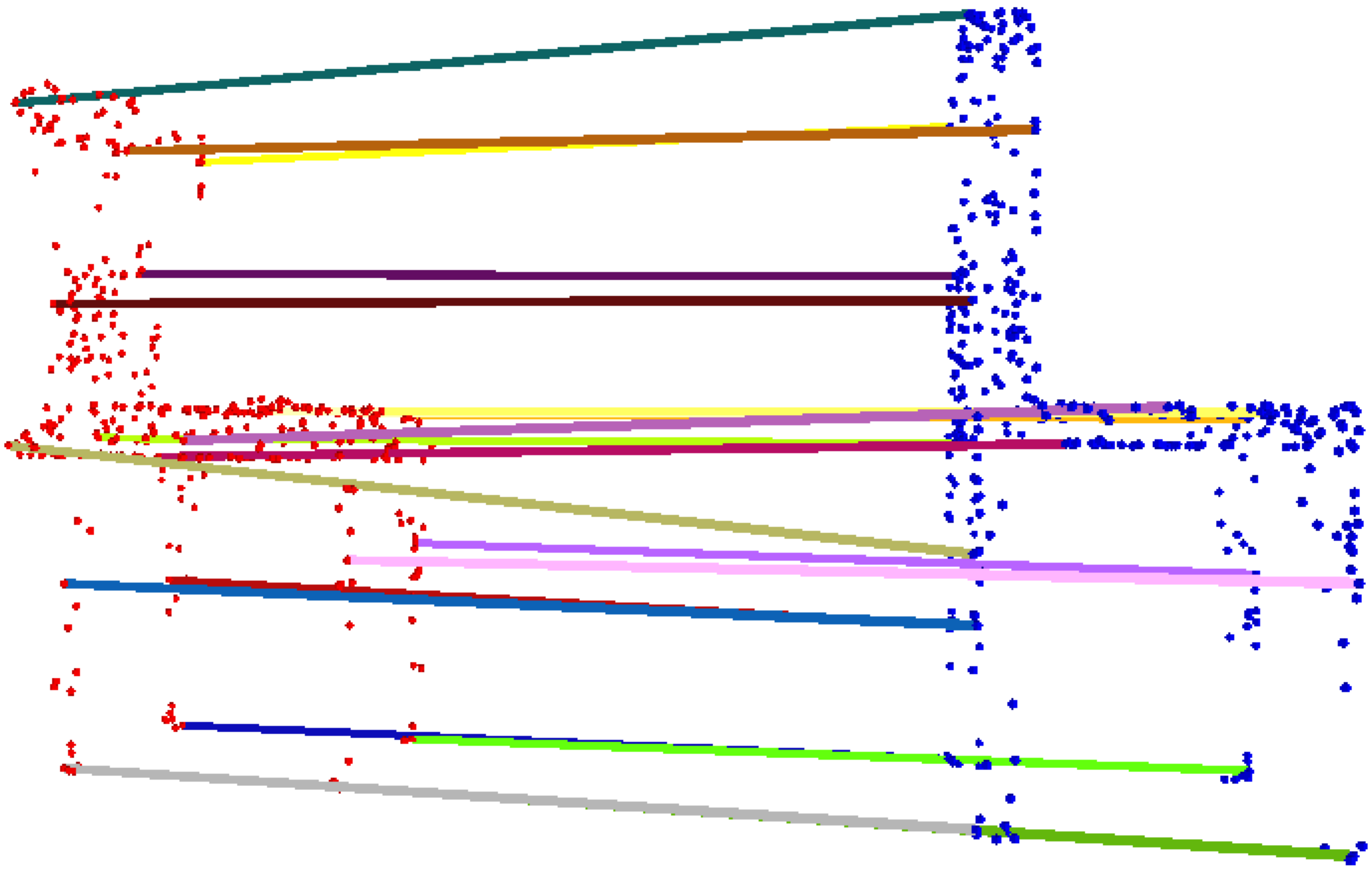


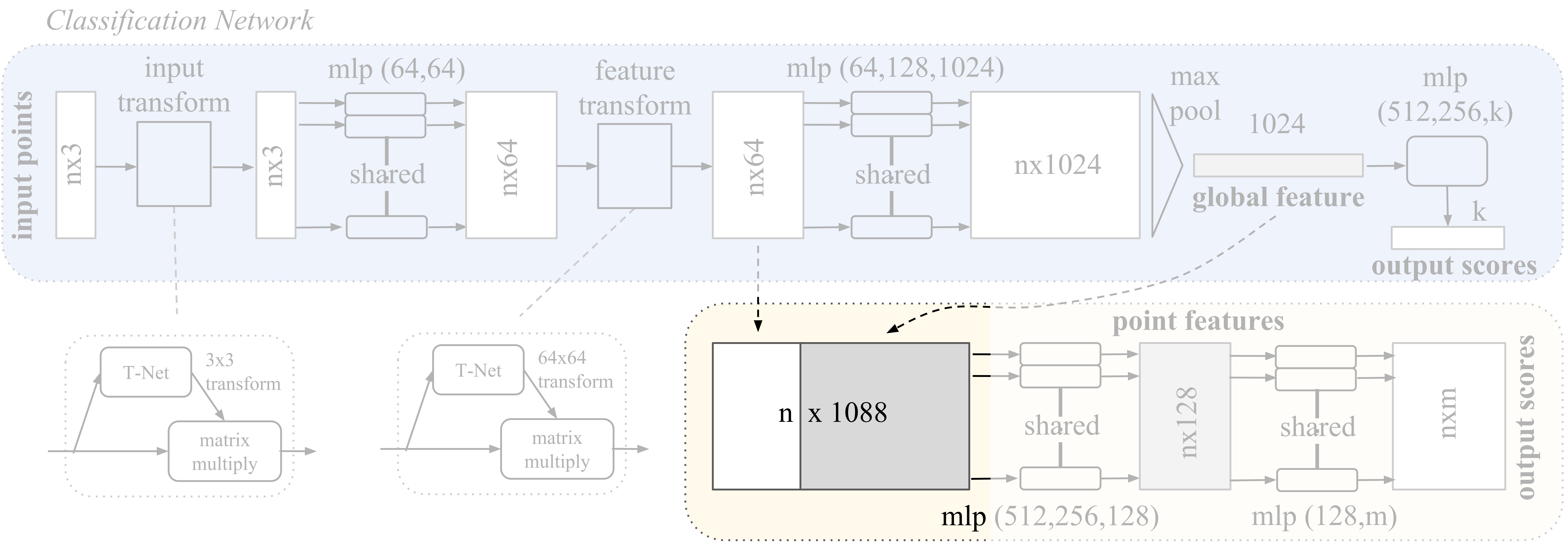
max of all points for all 1024 features
Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
order invariant!
point features
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

network to reduce
features to k class scores

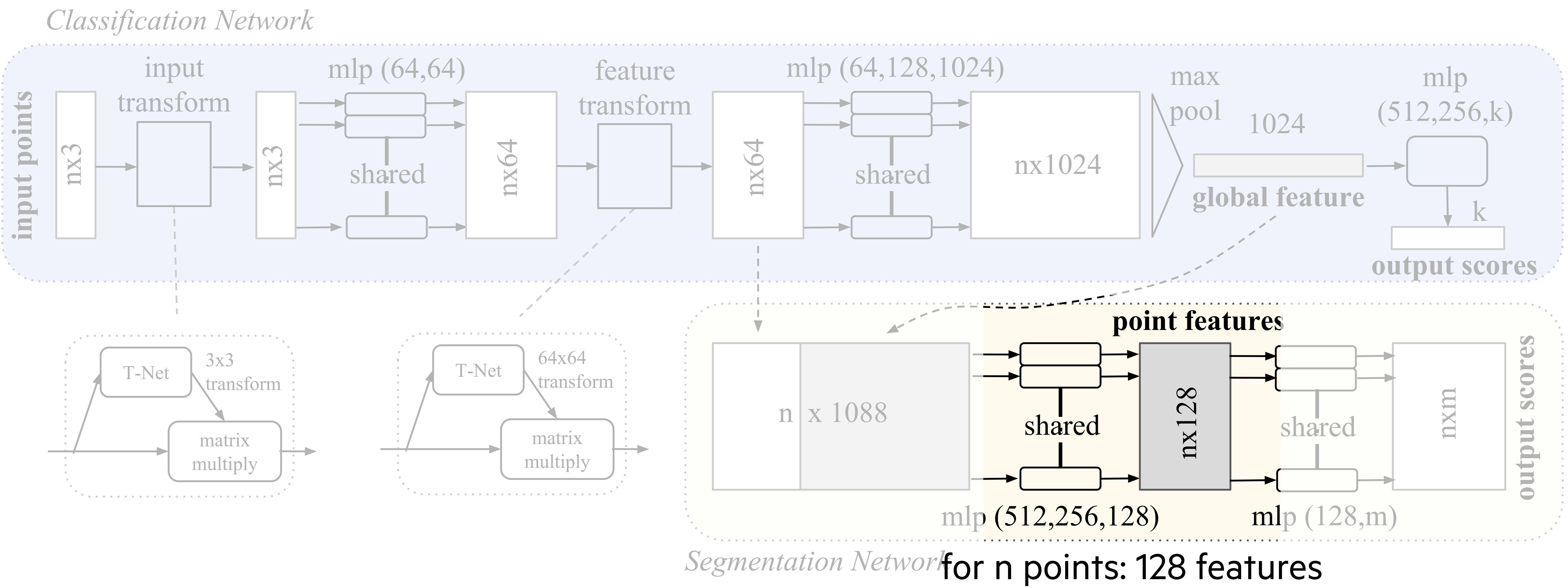



Query
Point Cloud
Top-5 Retrieval CAD Models

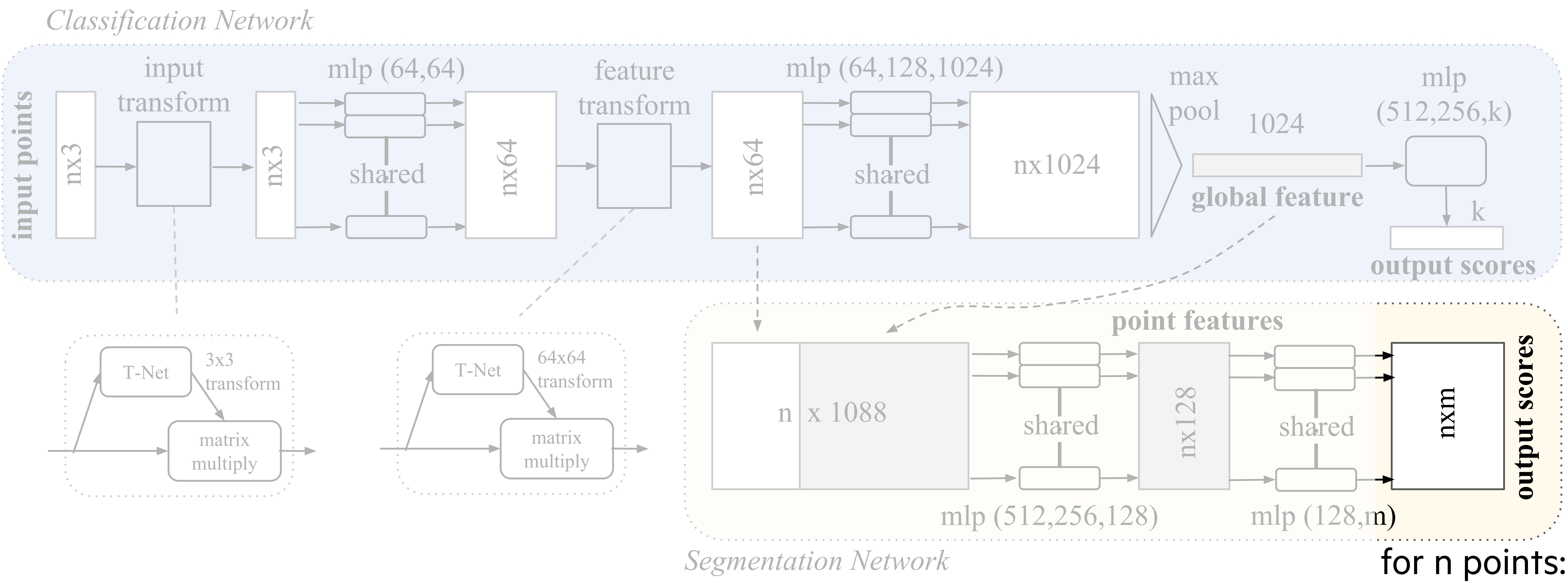




for n points: local + global features

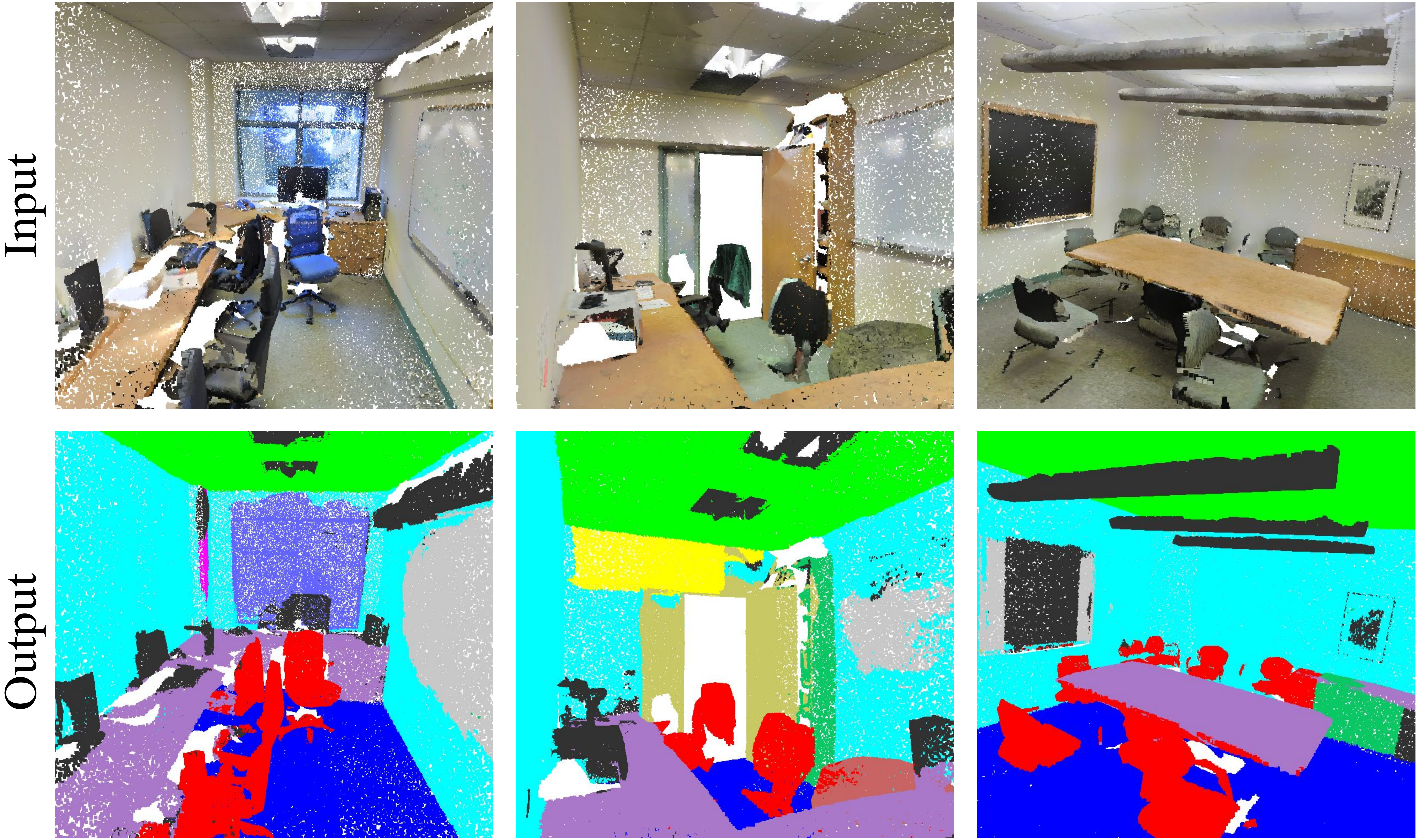


for n points: 128 features

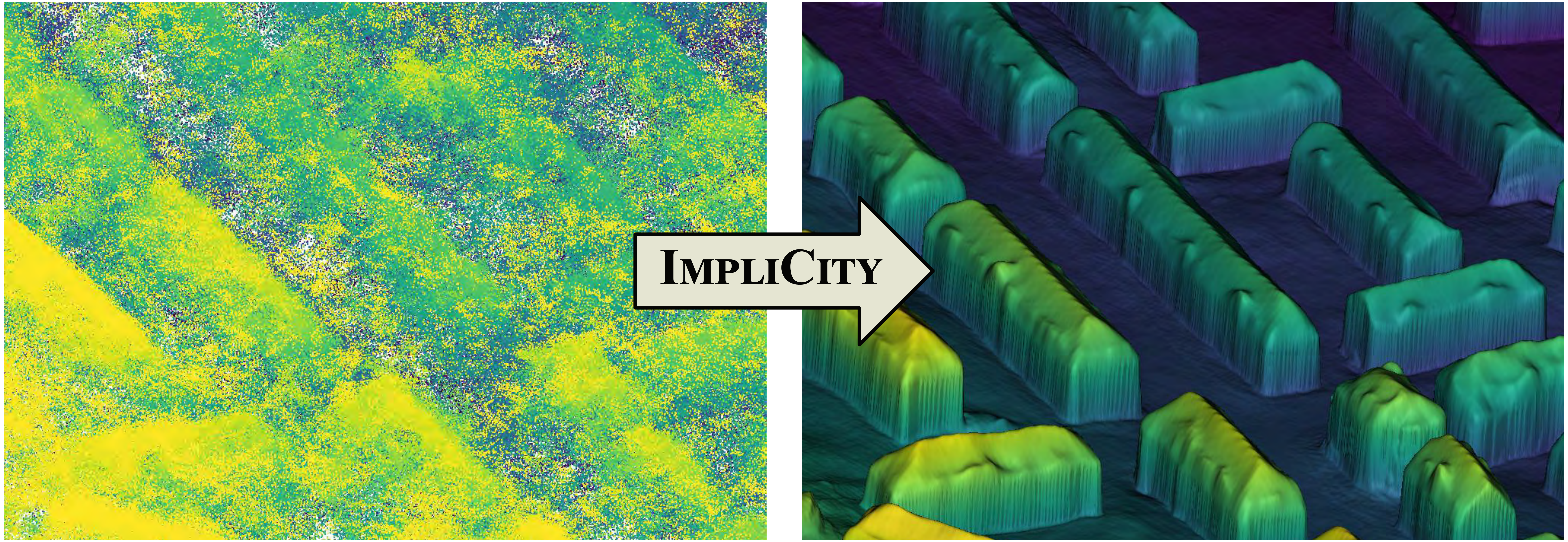


for n points:

m class scores

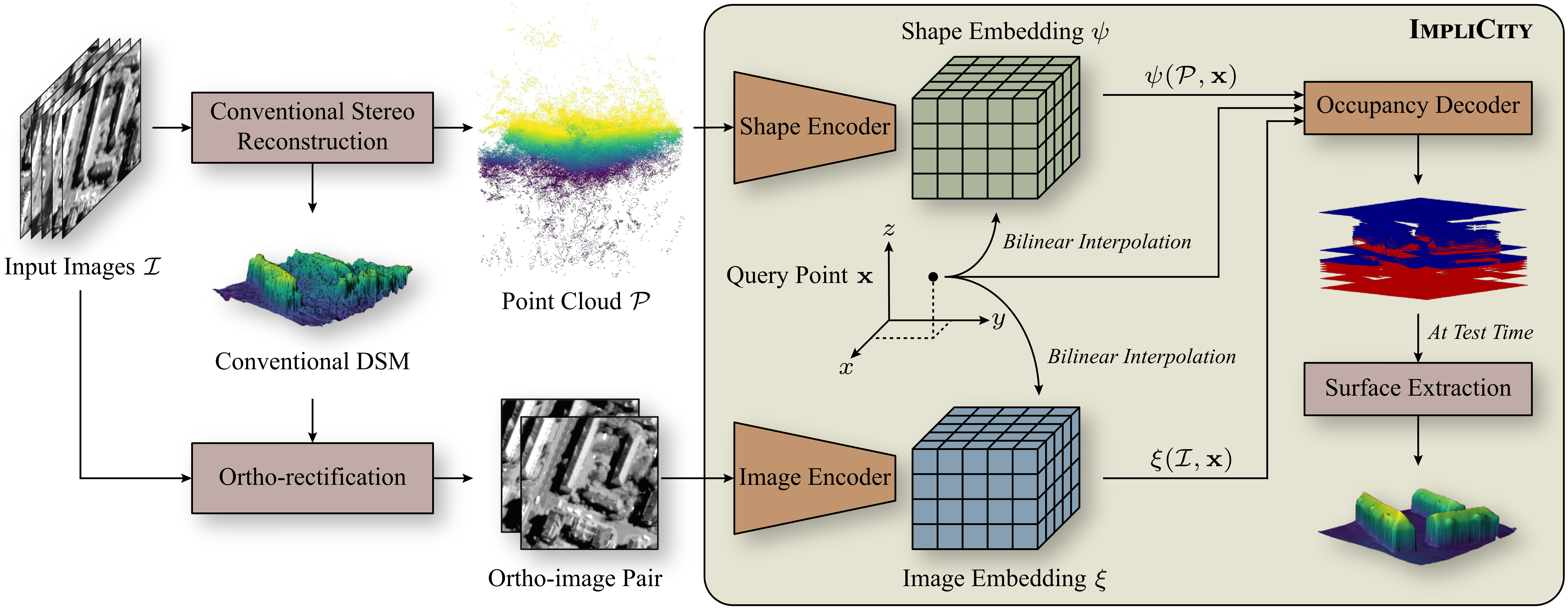

Input
Output

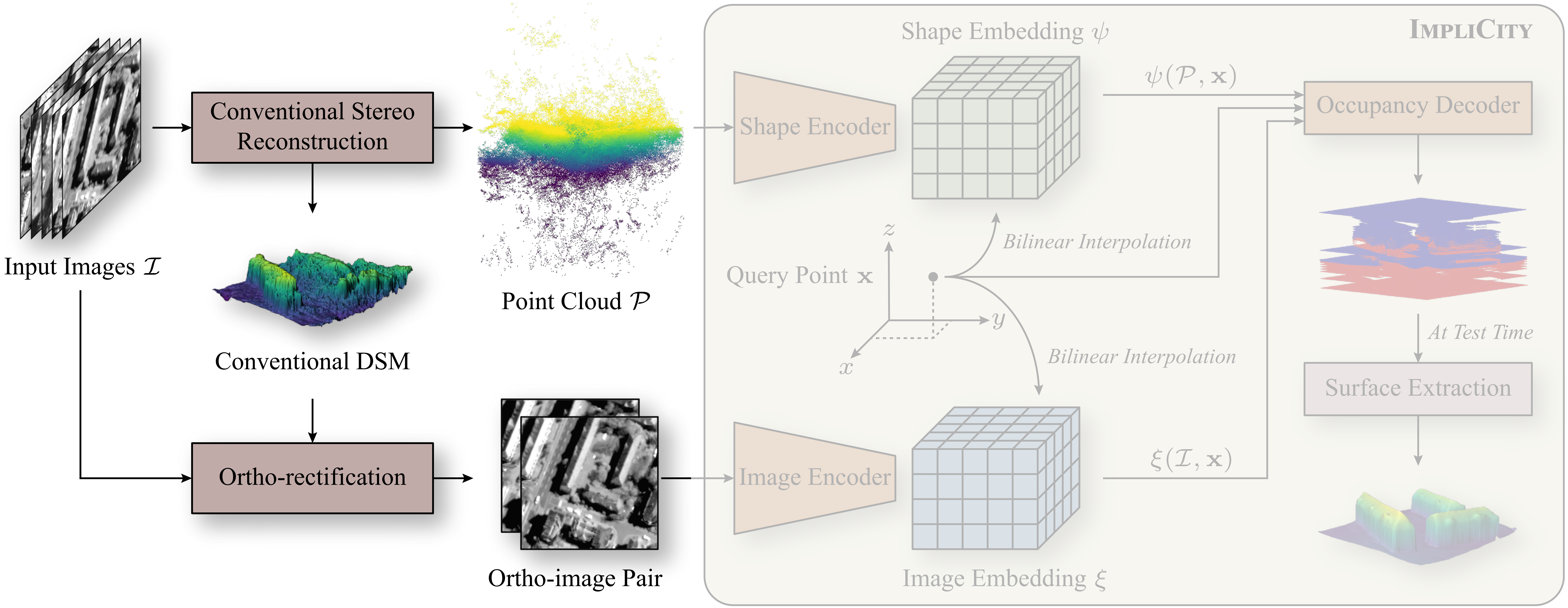
ImpliCity: City modeling from satellite images with deep occupancy fields



PC from satellite imagery

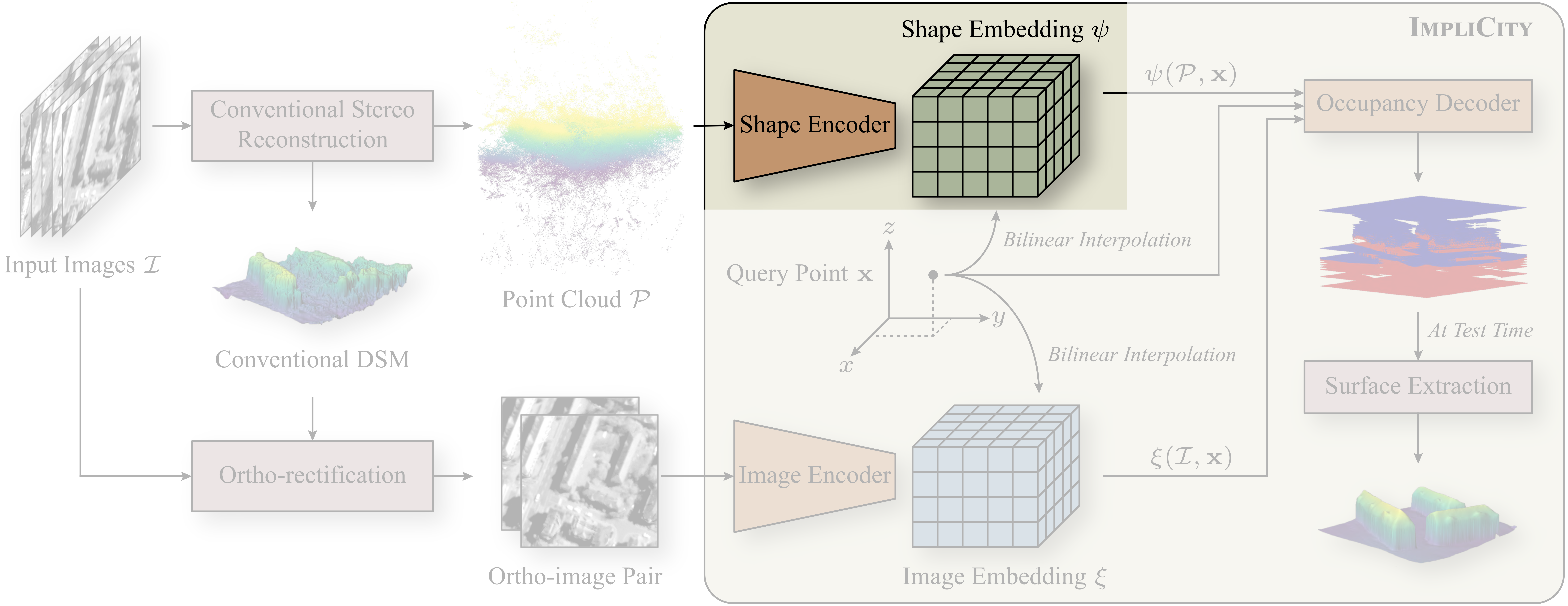
DSM

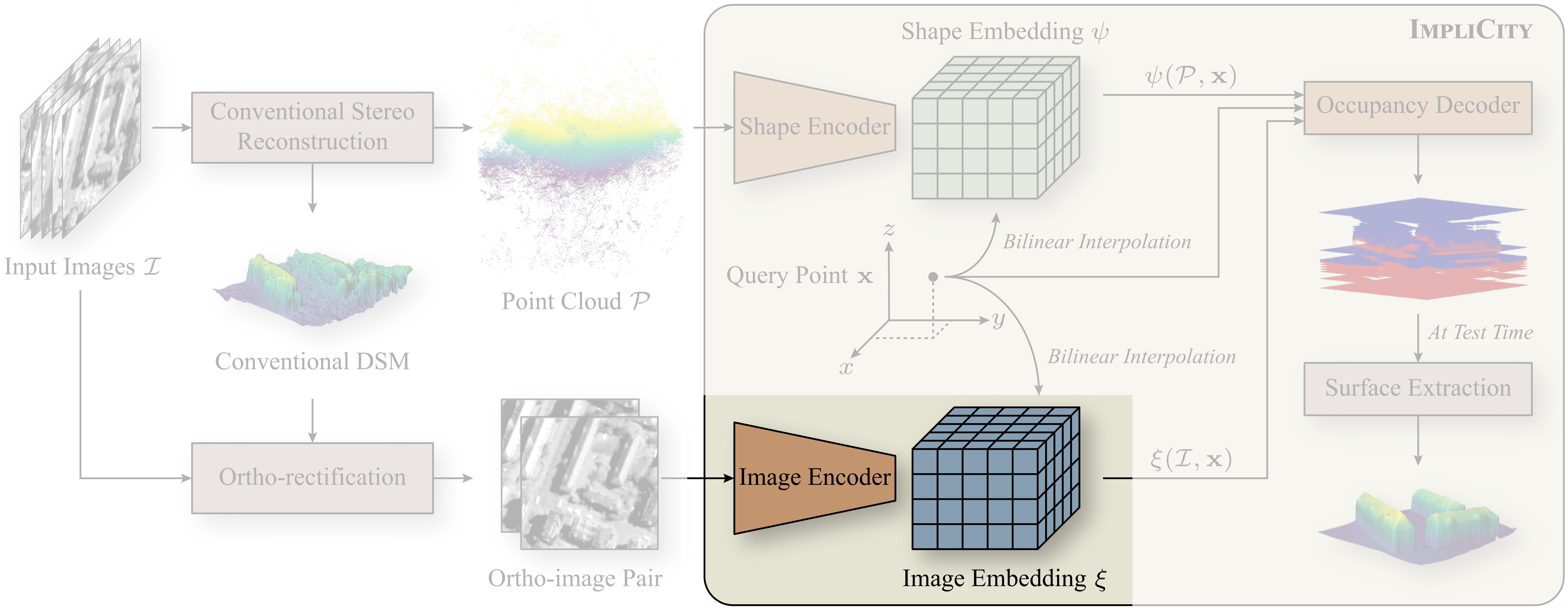

Input Images $\mathcal{I}$
Conventional Stereo Reconstruction
Conventional DSM
Ortho-rectification
Point Cloud $\mathcal{P}$
Ortho-image Pair
IMPLICITY
Shape Embedding ψ
Shape Encoder
$\psi(\mathcal{P}, \mathbf{x})$
Occupancy Decoder
Bilinear Interpolation
Query Point $\mathbf{x}$
z
y
x
Bilinear Interpolation
At Test Time
Surface Extraction
Image Encoder
$\xi(\mathcal{I}, \mathbf{x})$
Image Embedding ξ



traditional PC creation, DSM reconstruction & orthorectification

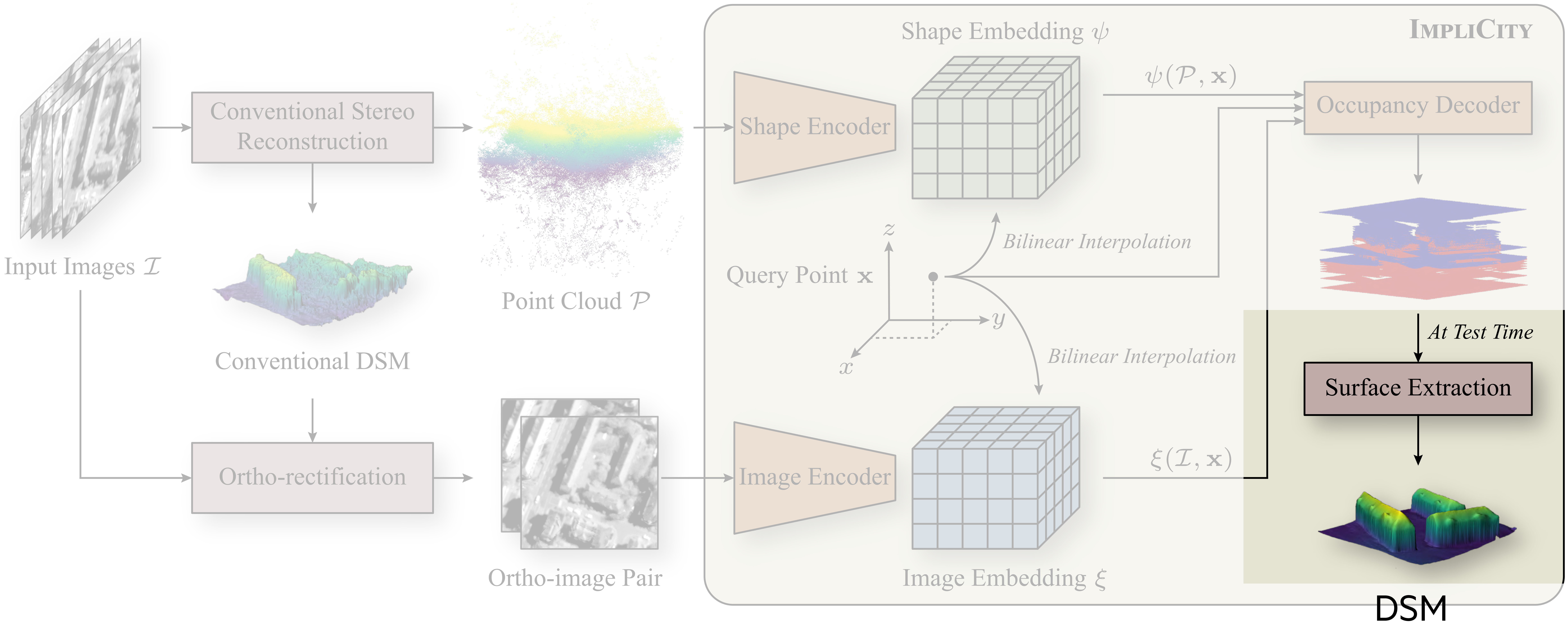
network to compute point features
(based on PointNet)





network to compute pixel features
(based on PointNet)

probability of occupancy
ImpliCity
Input Images $\mathcal{I}$
Conventional Stereo Reconstruction
Conventional DSM
Point Cloud $\mathcal{P}$
Ortho-rectification
Ortho-image Pair
Shape Embedding ψ
Shape Encoder
$\psi(\mathcal{P}, \mathbf{x})$
Occupancy Decoder
Bilinear Interpolation
Query Point $\mathbf{x}$
Bilinear Interpolation
At Test Time
Surface Extraction
Image Encoder
$\xi(\mathcal{I}, \mathbf{x})$
Image Embedding ξ



Input Images $\mathcal{I}$
Conventional Stereo Reconstruction
Conventional DSM
Point Cloud $\mathcal{P}$
Ortho-rectification
Ortho-image Pair
ImpliCity
Shape Embedding ψ
Shape Encoder
$\psi(\mathcal{P}, \mathbf{x})$
Occupancy Decoder
Bilinear Interpolation
Query Point $\mathbf{x}$
z
y
x
Bilinear Interpolation
At Test Time
Surface Extraction
Image Encoder
$\xi(\mathcal{I}, \mathbf{x})$
Image Embedding ξ
DSM


Conventional DSM
IMPLICITY-stereo (ours)
Google Earth view

Some thoughts on ML/DL with 3D

What to do next?

1. Today:
    - Start with Homework 3 (BIM to Geo using voxels)
    - Study for final exam (Lessons 1.1-6.2)
2. Wednesday: overview applications (1st hour) and help (2nd hour)
3. Thursday: help session with Dimitris

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

References

- Anna Labetski, Stelios Vitalis, Filip Biljecki, Ken Arroyo Ohori & Jantien Stoter (2023). 3D building metrics for urban morphology, International Journal of Geographical Information Science, 37:1, 36-67.

- Charles R. Qi, Hao Su, Kaichun Mo & Leonidas J. Guibas (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, CVPR 2017.

- Max Jaderberg, Karen Simonyan, Andrew Zisserman & Koray Kavukcuoglu (2015). Spatial Transformer Networks, Advances in Neural Information Processing Systems 28.

- Corinne Stucker, Bingxin Ke, Yuanwen Yue, Shengyu Huang, Iro Armeni & Konrad Schindler (2022). ImpliCity: City Modeling from Satellite Images with Deep Implicit Occupancy Fields, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-2-2022, 193-201.