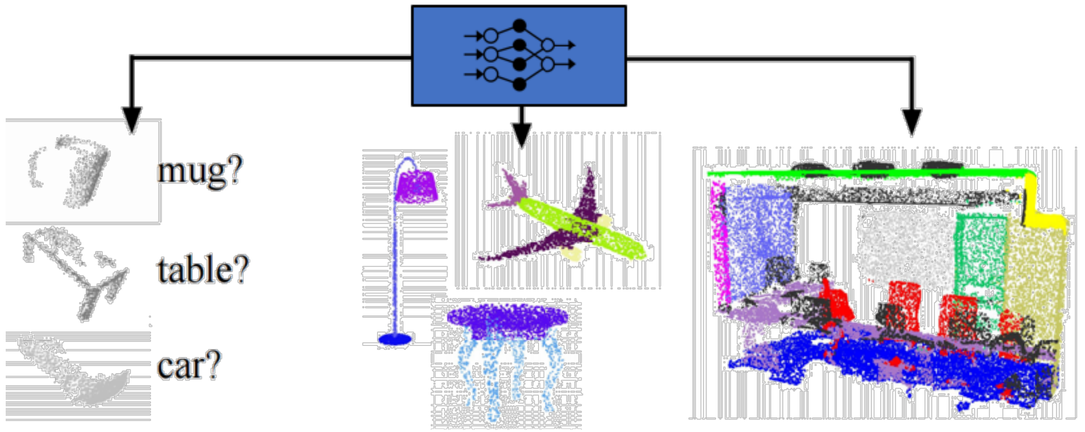


14. Előadás: 3D Mélytanulás

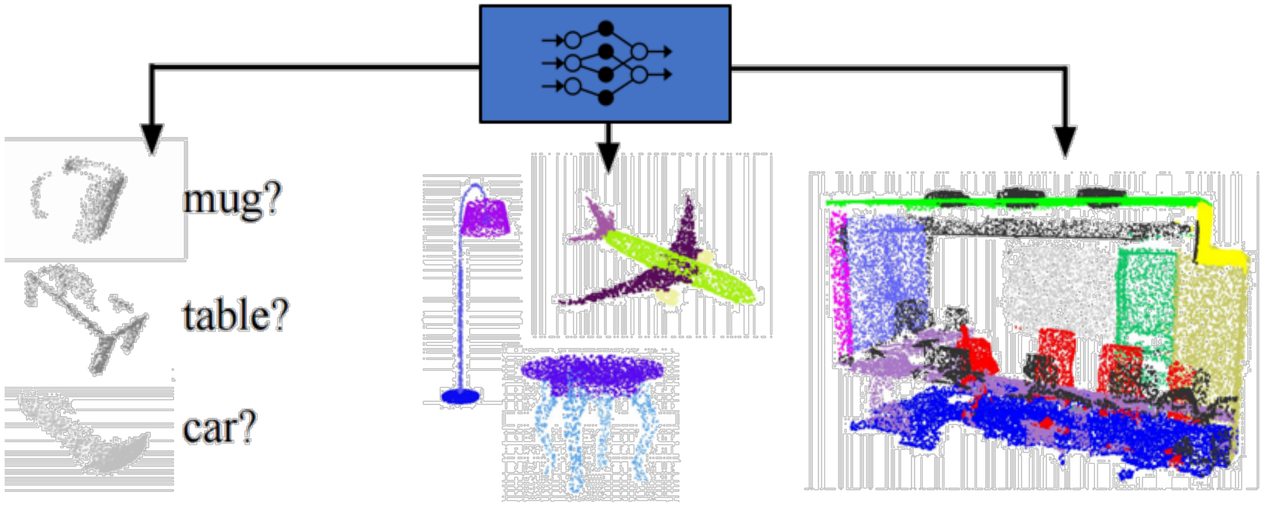
Generatív AI és Inverz Módszerek a Képszintézisben

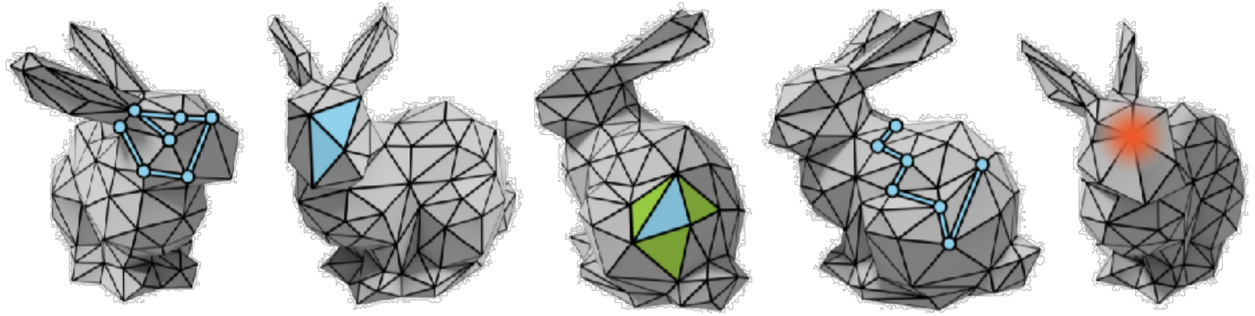
BME-VIK IIT, 2026

Dr. Vaitkus Márton

3D Mélytanulás

- Mélytanulás 3D adatokon — hogyan?
- Kihívások:
  - Irreguláris adatszerkezetek
  - Adatok mennyisége és minősége
  - “Nyers” adatok (pl. pontfelhők) mérete, tisztasága
  - Az adatszerkezet nem választható szét a benne tárolt információtól (Hány pont? Milyen adatív rács? Milyen topológiájú háló?)





3D Mélytanulás

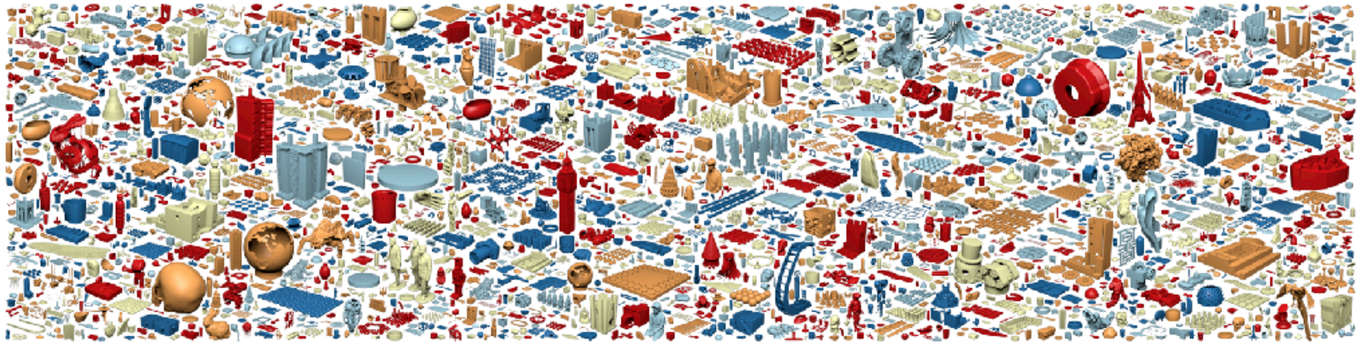
3D adathalmazok

The Stanford 3D Scanning Repository

Large Geometric Models Archive

Georgia Institute of Technology

http://redwood-data.org/3dscan/

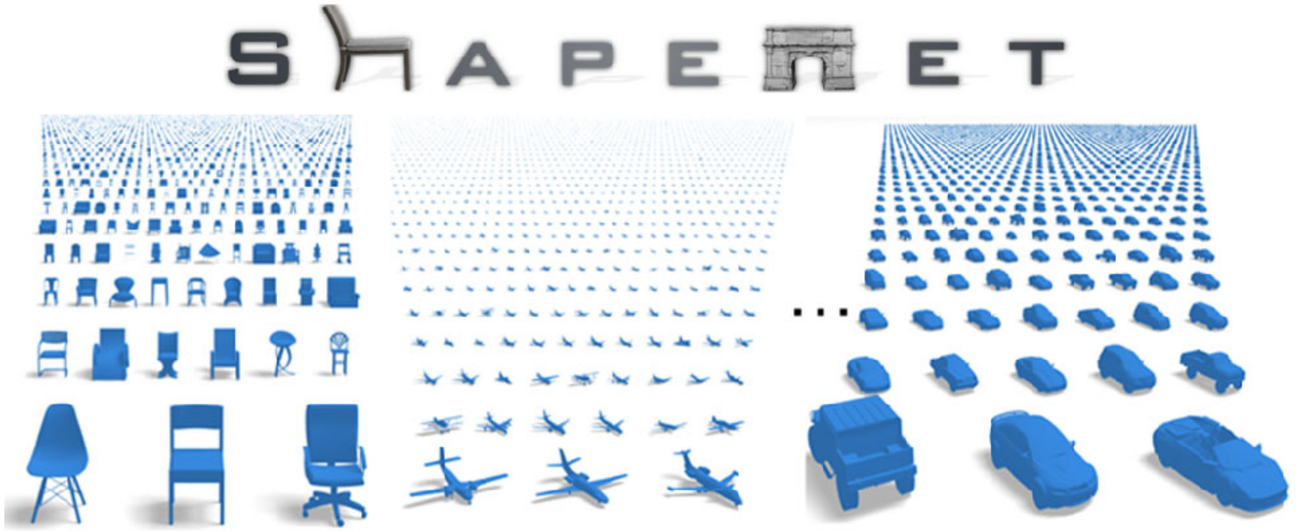


Thingi10K: A Dataset of 10,000 3D-Printing Models

3D Mélytanulás

3D adathalmazok

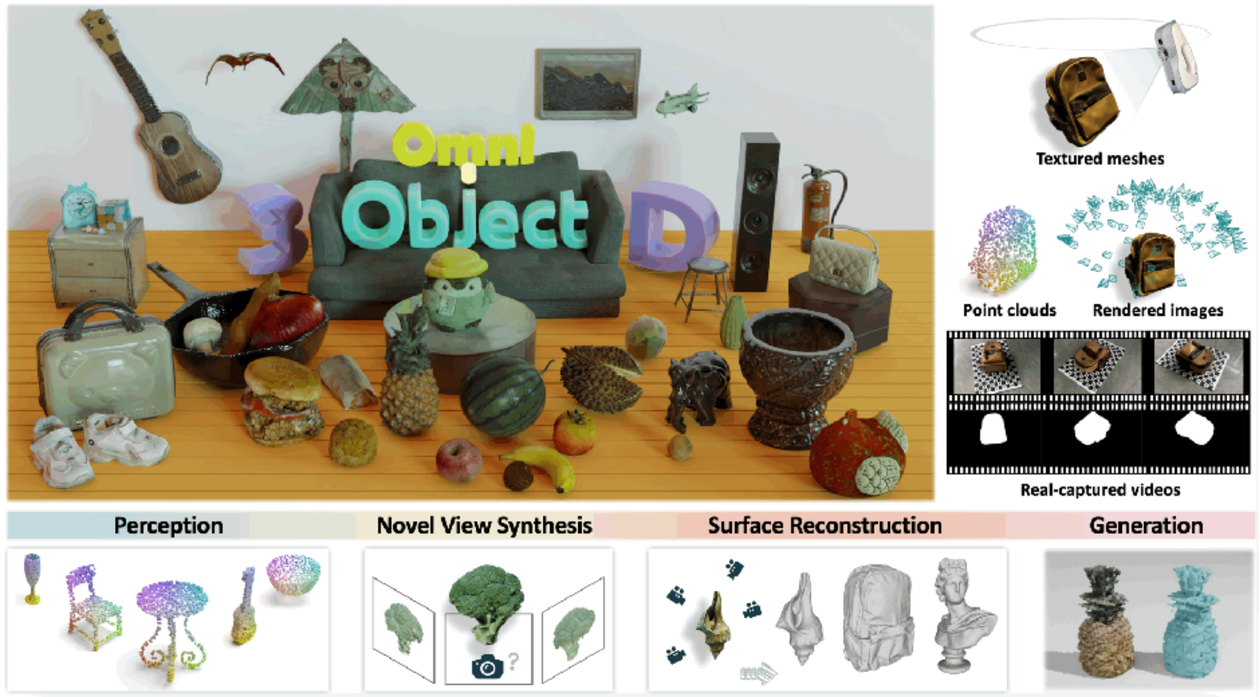
12000+ 3D modell, 40 kategória



Összesen 3M 3D modell
300k 3D modell 3000+ kategóriába sorolva
ShapeNetCore: 50k “tiszta” modell, 55 kategória

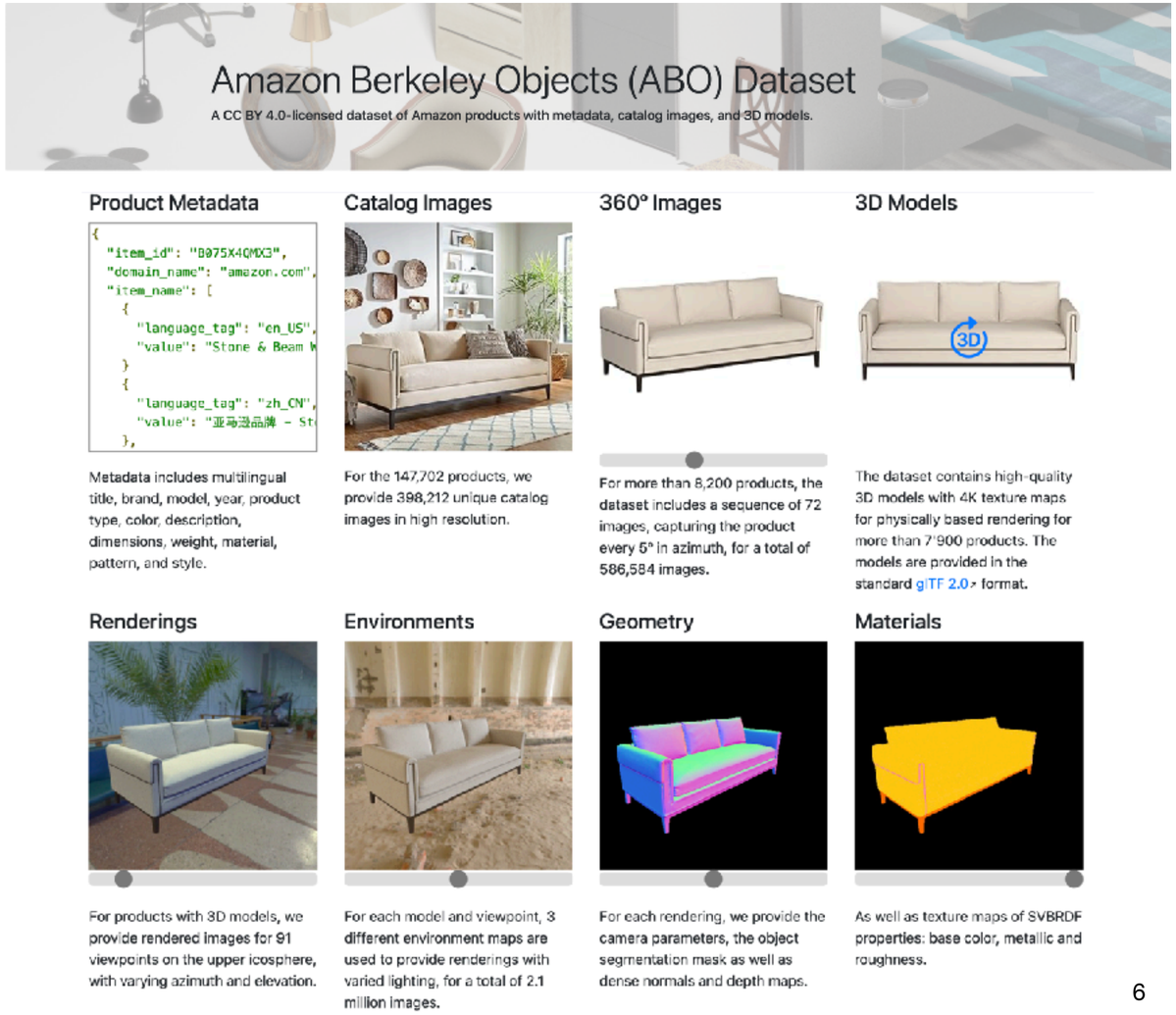
3D Mélytanulás

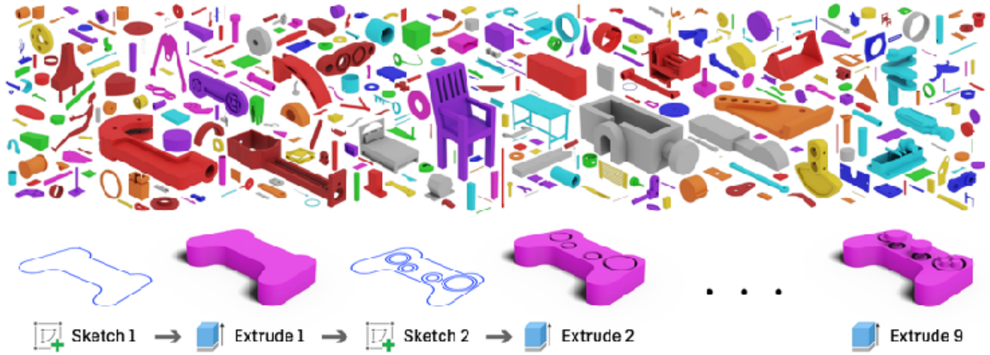
3D adathalmazok



3D Mélytanulás

3D adathalmazok — CAD modellek



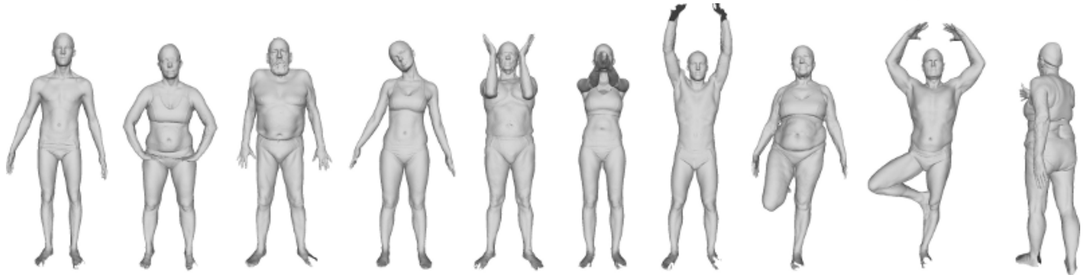


Fusion360Gallery

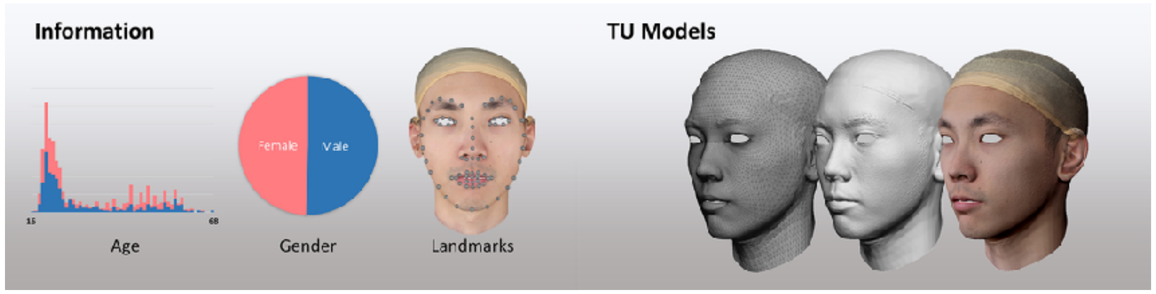
3D Mélytanulás

3D adathalmazok — Humán modellek

FAUST: Dataset and evaluation for 3D mesh registration



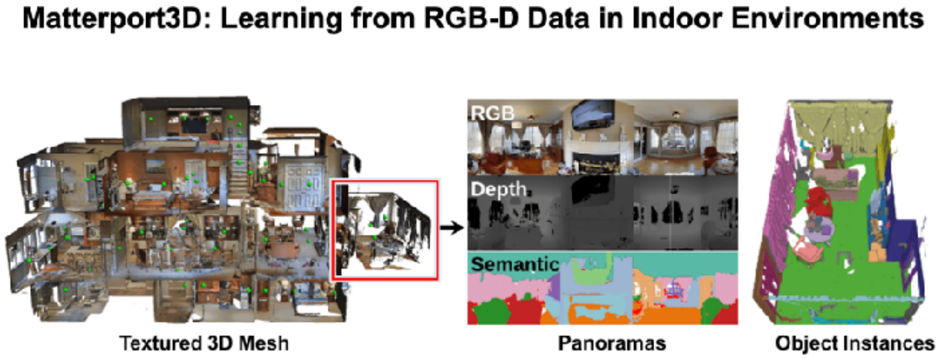
FACESCAPE

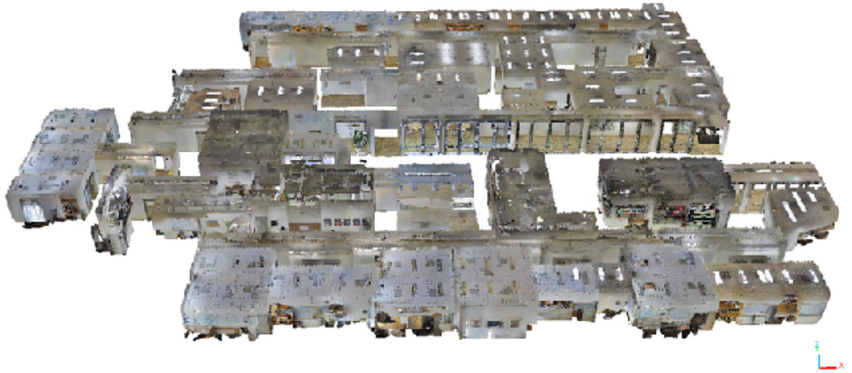


Renderpeople®

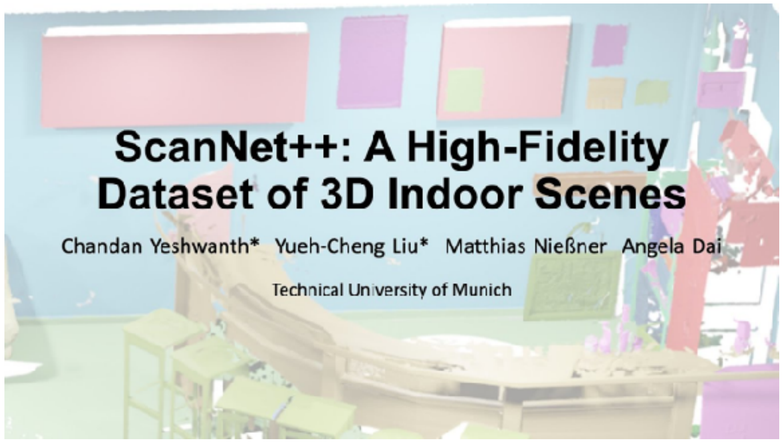
3D Mélytanulás

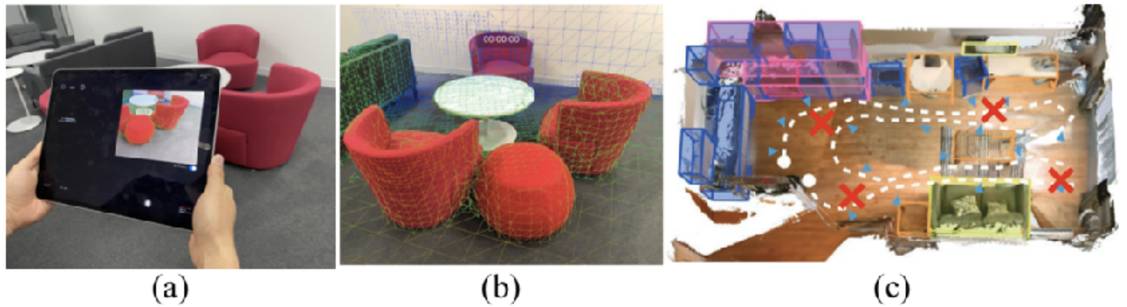
3D adathalmazok — Beltéri





Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS)

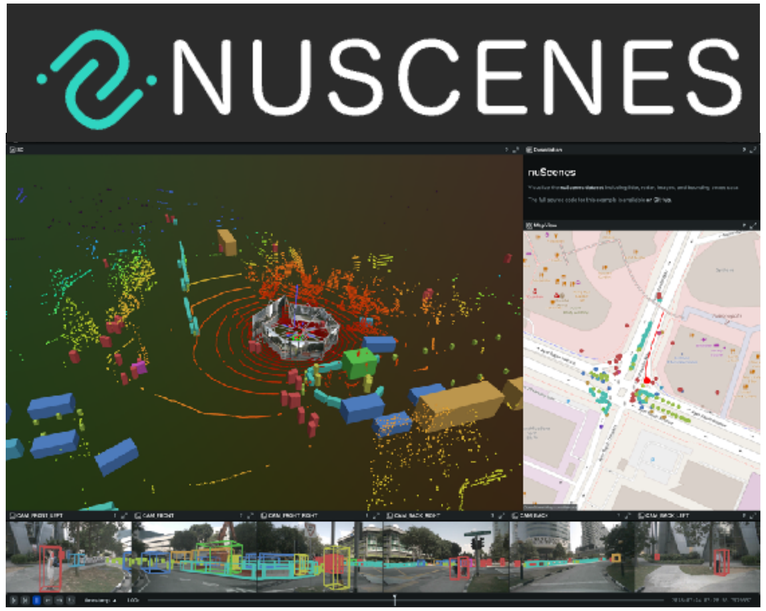


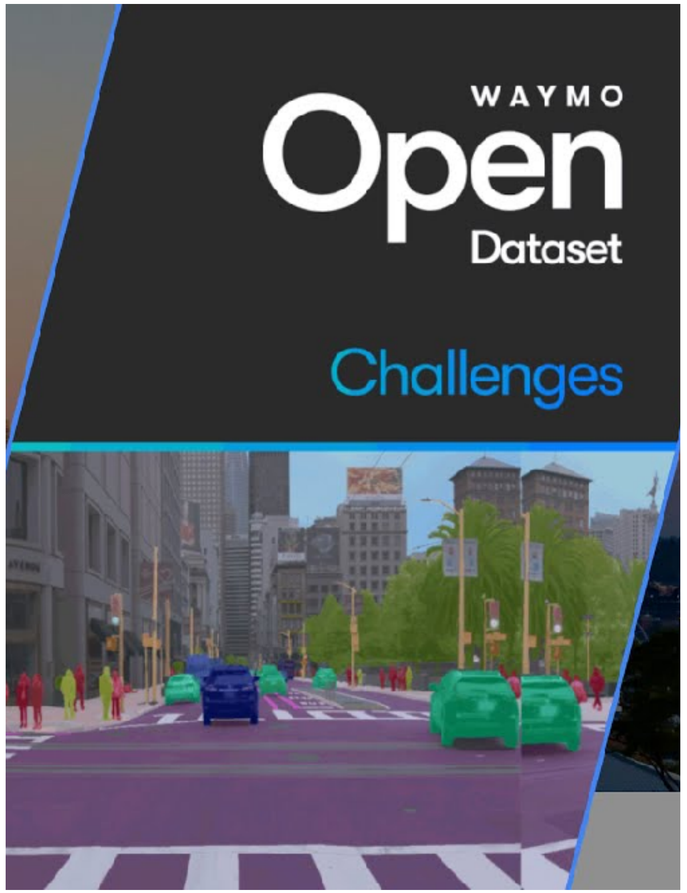


Apple ARKitScenes

3D Mélytanulás

3D adathalmazok — Kültéri





3D Mélytanulás

3D adathalmazok — Procedurális generálás

```
>>> import infinigen
>>> infinigen.generate()
>>>
```

https://infinigen.org/

3D Mélytanulás

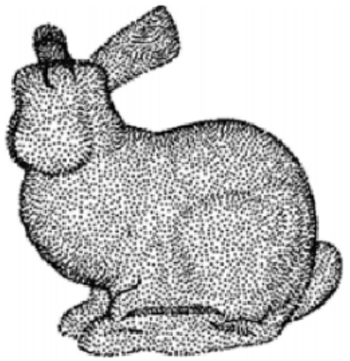
3D adathalmazok — Procedurális generálás

```
>>> import infinigen
>>> infinigen.generate()
>>>
```

https://infinigen.org/

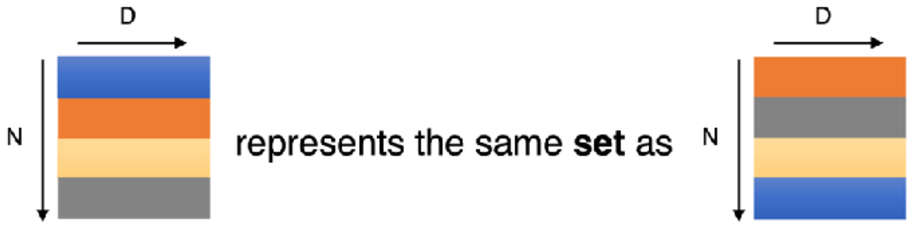
3D Mélytanulás

Pontfelhők — PointNet



$$\{p_i = (x_1, x_2, \ldots x_D),\ i = 1,\ldots N\}$$

Neurális háló pontfelhőkre?



Model needs to be invariant to N! permutations

Symmetric functions:

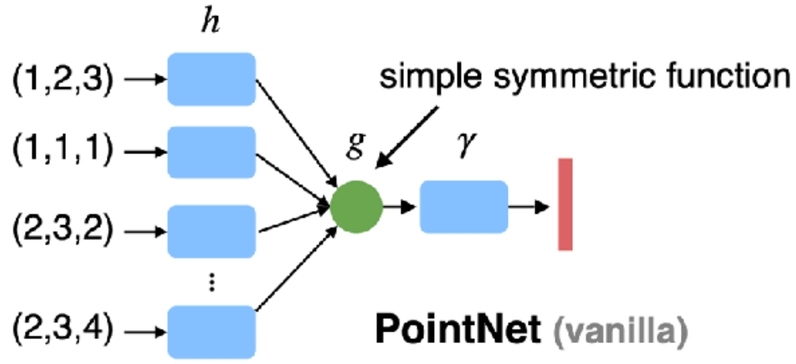
$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

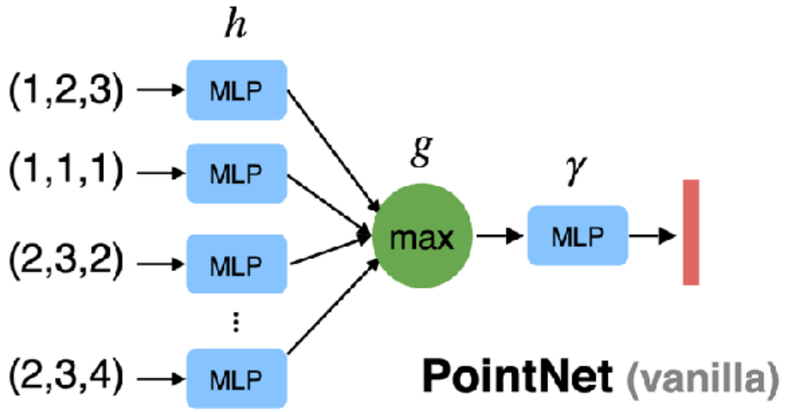
$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

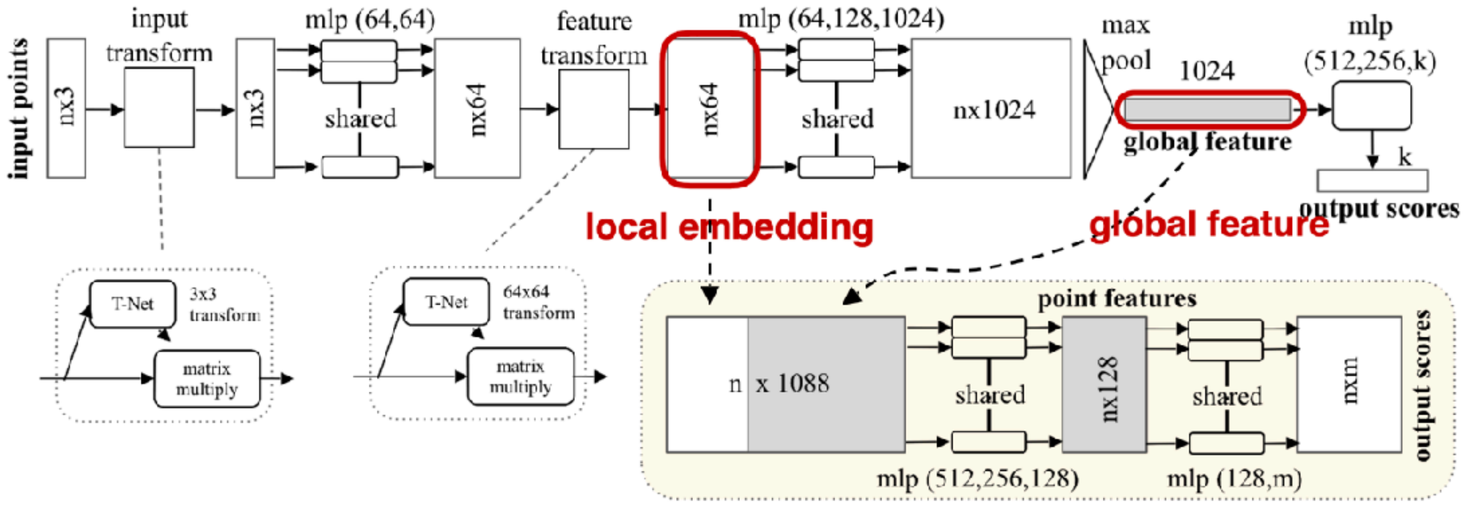
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



3D Mélytanulás

Pontfelhők — PointNet — Architektúra





3D Mélytanulás

Pontfelhők — PointNet — Architektúra

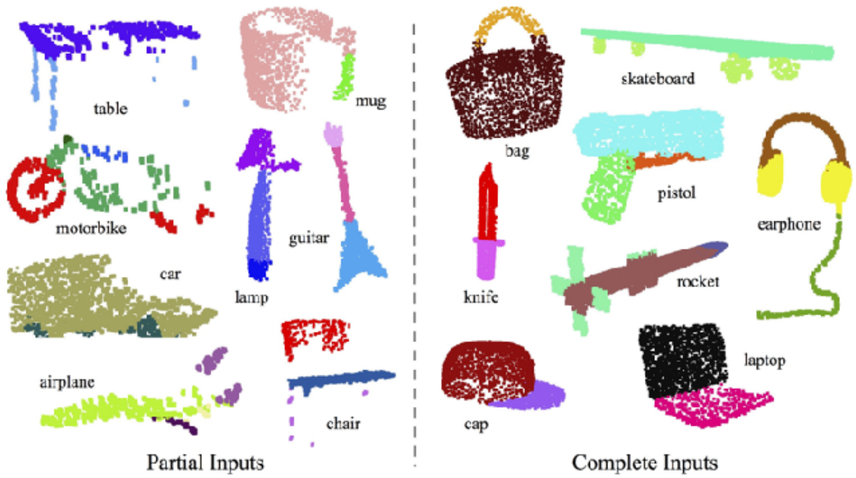
```python
class PointNetEncoder(nn.Module):
    def __init__(self, global_feat=True, feature_transform=False, channel=3):
        super(PointNetEncoder, self).__init__()
        self.stn = STN3d(channel)
        self.conv1 = torch.nn.Conv1d(channel, 64, 1)
        self.conv2 = torch.nn.Conv1d(64, 128, 1)
        self.conv3 = torch.nn.Conv1d(128, 1024, 1)
        self.bn1 = nn.BatchNorm1d(64)
        self.bn2 = nn.BatchNorm1d(128)
        self.bn3 = nn.BatchNorm1d(1024)
        self.global_feat = global_feat
        self.feature_transform = feature_transform
        if self.feature_transform:
            self.fstn = STNkd(k=64)
```

```python
def forward(self, x):
    B, D, N = x.size()
    trans = self.stn(x)
    x = x.transpose(2, 1)
    if D > 3:
        feature = x[:, :, 3:]
        x = x[:, :, :3]
    x = torch.bmm(x, trans)
    if D > 3:
        x = torch.cat([x, feature], dim=2)
    x = x.transpose(2, 1)
    x = F.relu(self.bn1(self.conv1(x)))

    if self.feature_transform:
        trans_feat = self.fstn(x)
        x = x.transpose(2, 1)
        x = torch.bmm(x, trans_feat)
        x = x.transpose(2, 1)
    else:
        trans_feat = None

    pointfeat = x
    x = F.relu(self.bn2(self.conv2(x)))
    x = self.bn3(self.conv3(x))
    x = torch.max(x, 2, keepdim=True)[0]
    x = x.view(-1, 1024)
    if self.global_feat:
        return x, trans, trans_feat
    else:
        x = x.view(-1, 1024, 1).repeat(1, 1, N)
        return torch.cat([x, pointfeat], 1), trans, trans_feat
```

3D Mélytanulás

Pontfelhők — PointNet

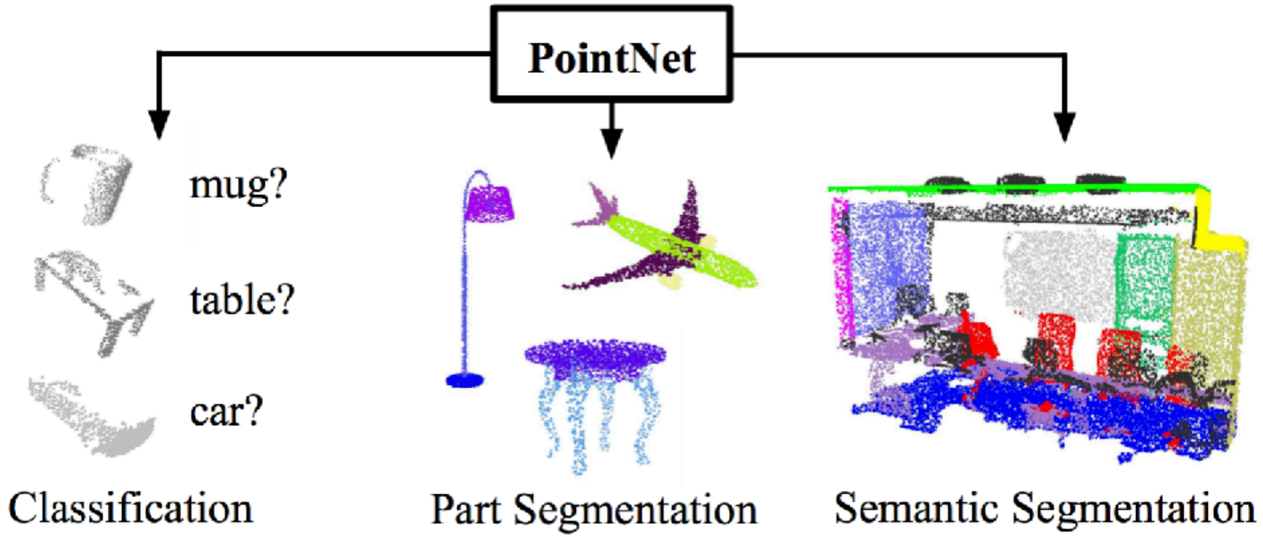
	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	**89.2**
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**



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

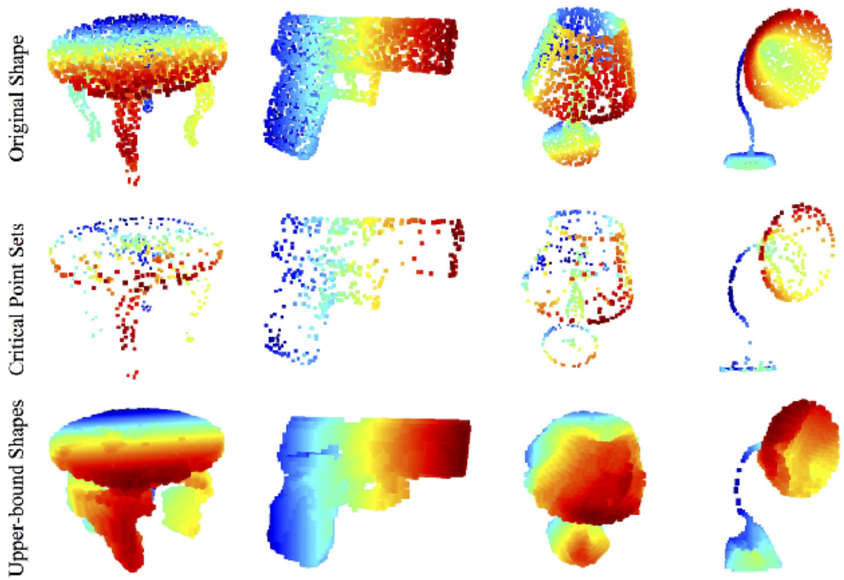
Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas
Stanford University

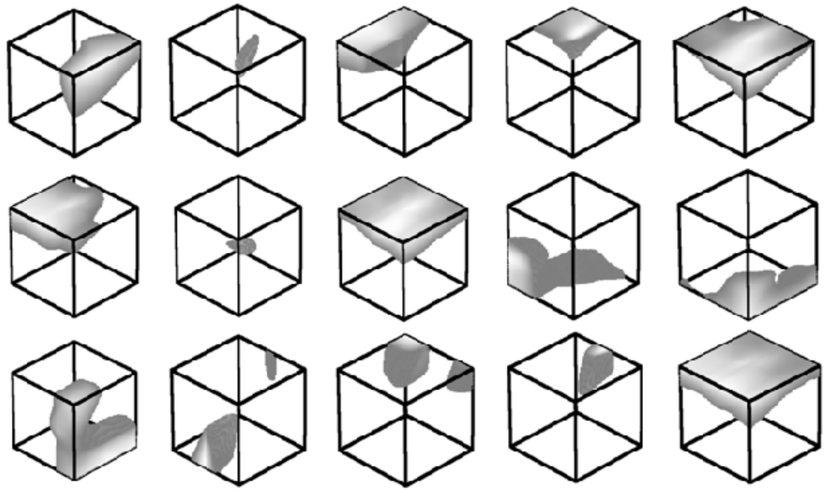
Cited by 22063

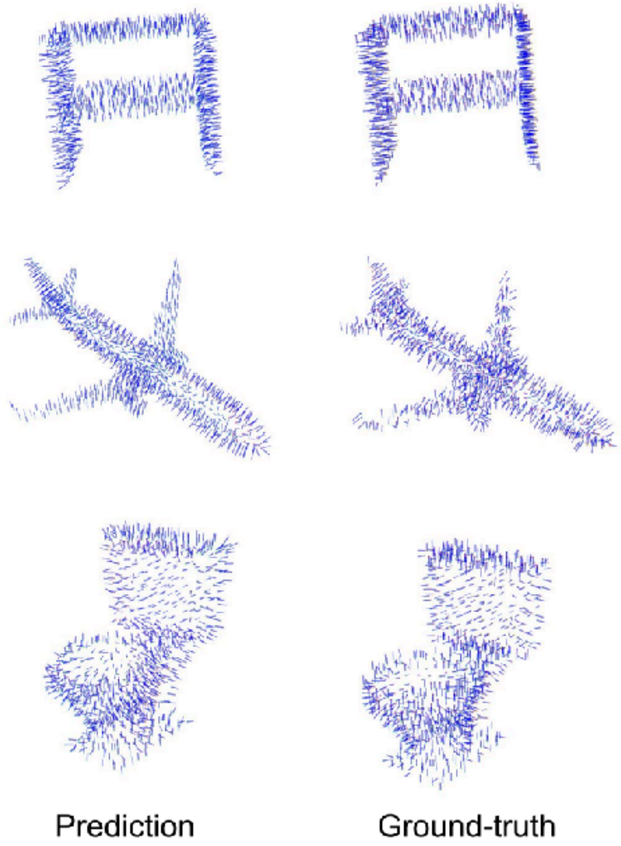


3D Mélytanulás

Pontfelhők — PointNet — Mit tanul meg?





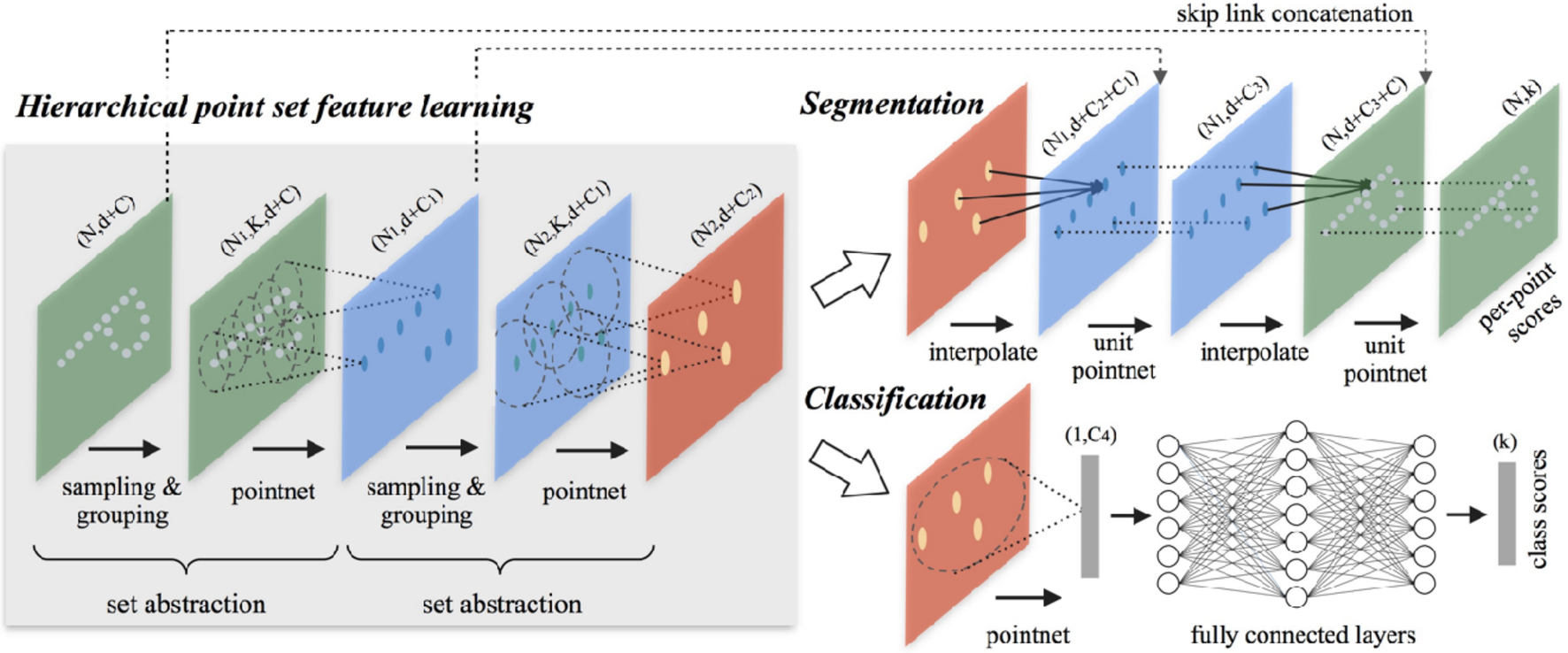


3D Mélytanulás

Pontfelhők — PointNet++

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

Charles R. Qi Li Yi Hao Su Leonidas J. Guibas
Stanford University



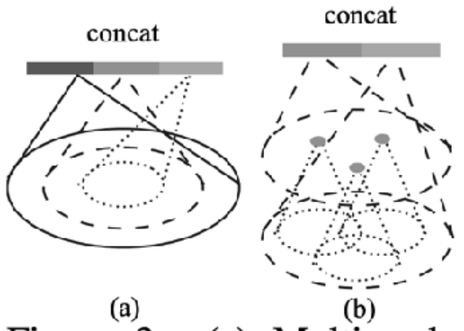


Figure 3: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

3D Mélytanulás

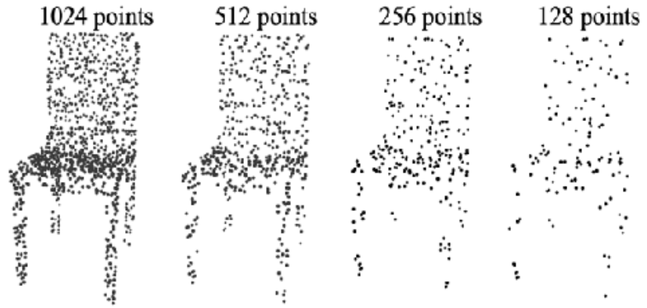
Pontfelhők — PointNet++

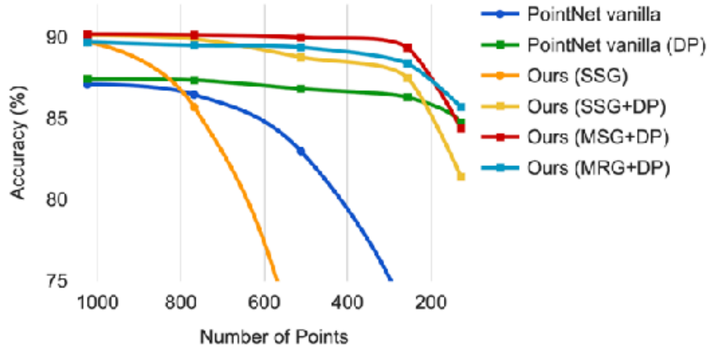
Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

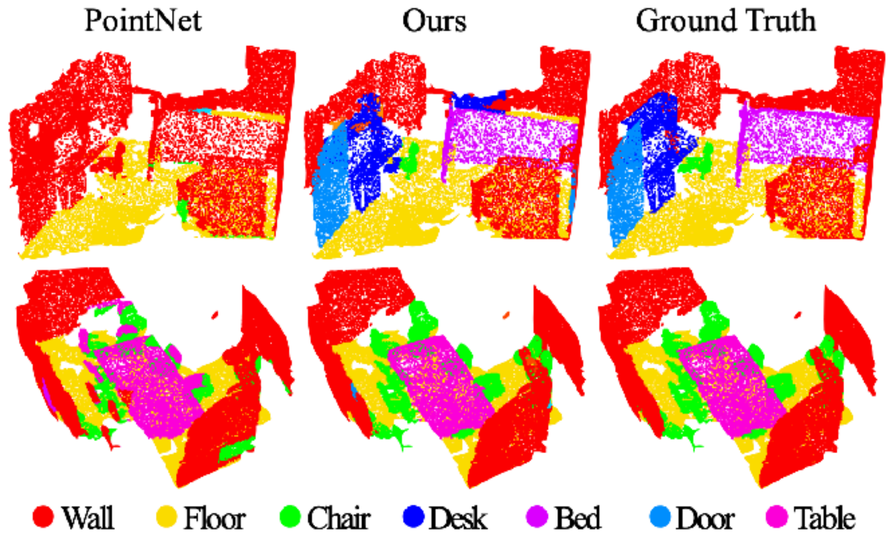
Table 1: MNIST digit classification.

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.

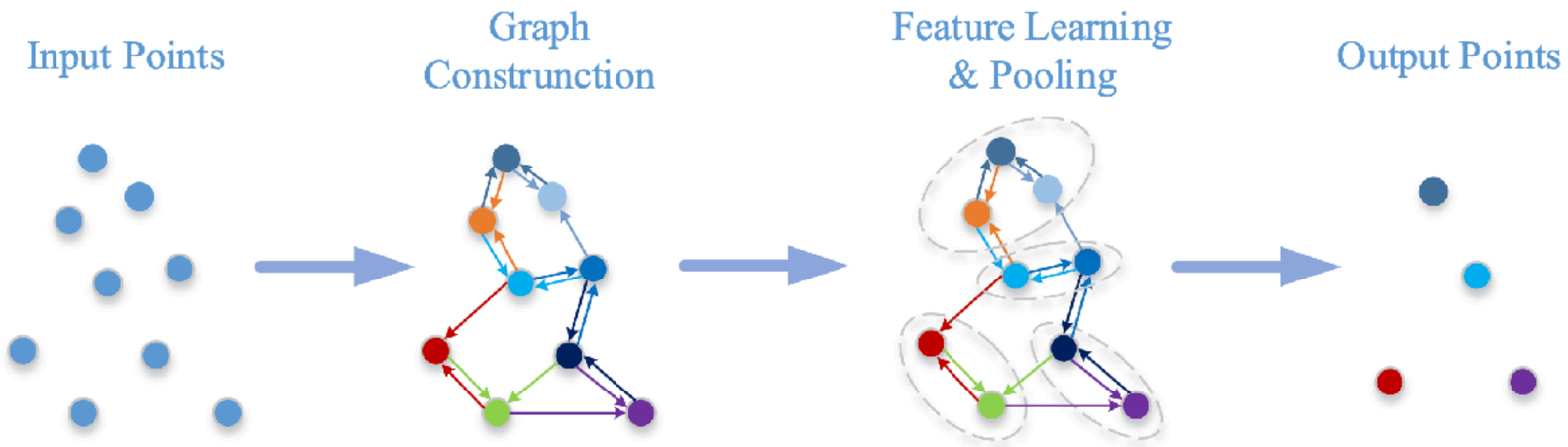






3D Mélytanulás

Pontfelhők — Gráf neurális hálók

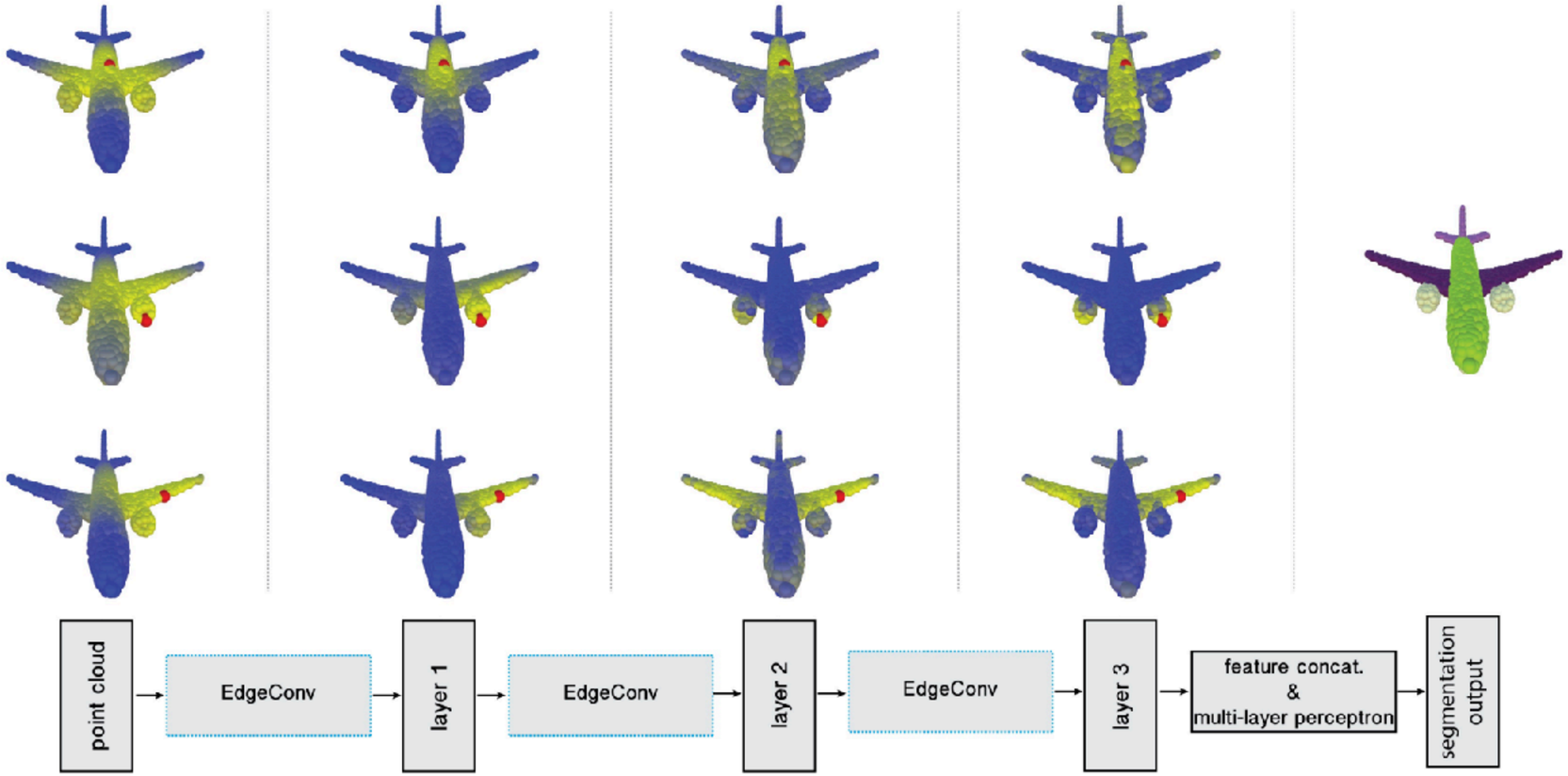


3D Mélytanulás

Pontfelhők — Dynamic Graph CNN

Dynamic Graph CNN for Learning on Point Clouds

YUE WANG, Massachusetts Institute of Technology
YONGBIN SUN, Massachusetts Institute of Technology
ZIWEI LIU, UC Berkeley / ICSI
SANJAY E. SARMA, Massachusetts Institute of Technology
MICHAEL M. BRONSTEIN, Imperial College London / USI Lugano
JUSTIN M. SOLOMON, Massachusetts Institute of Technology

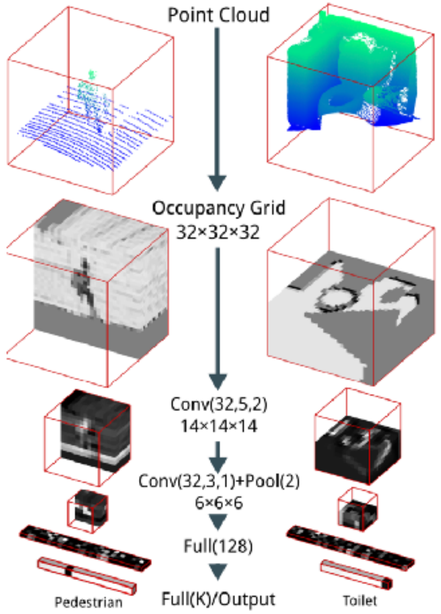


3D Mélytanulás

Pontfelhők — Voxel-alapú CNN

VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition

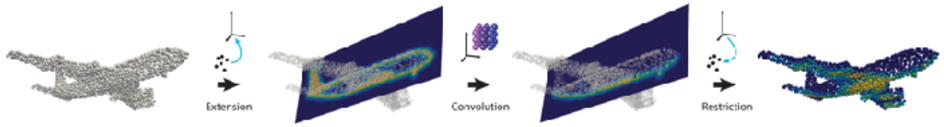
Daniel Maturana and Sebastian Scherer



Point Convolutional Neural Networks by Extension Operators

Matan Atzmon* Haggai Maron* Yaron Lipman

Weizmann Institute of Science



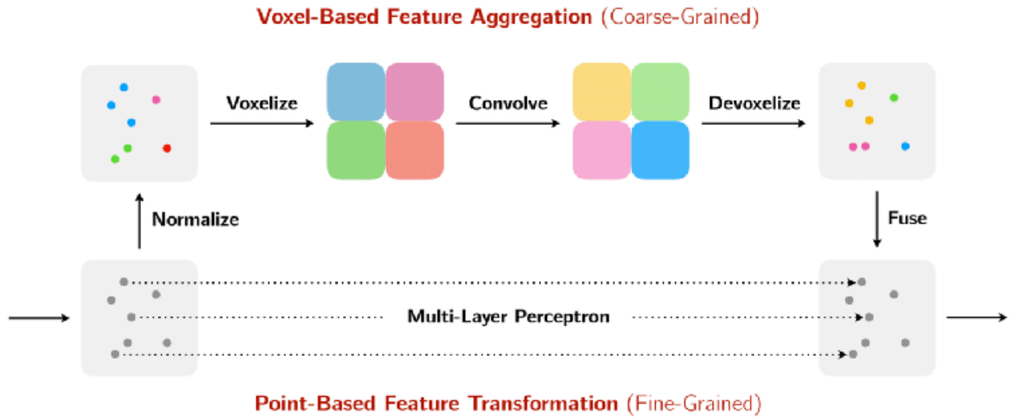
Point-Voxel CNN for Efficient 3D Deep Learning

Zhijian Liu* MIT

Haotian Tang* Shanghai Jiao Tong University

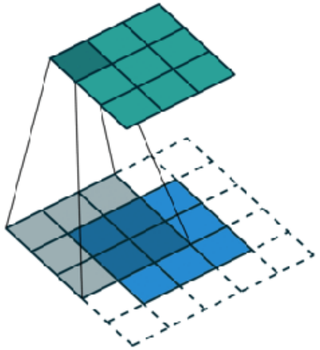
Yujun Lin MIT

Song Han MIT

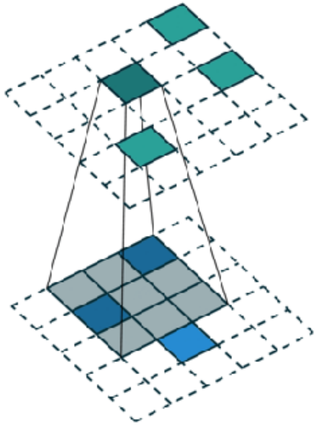


3D Mélytanulás

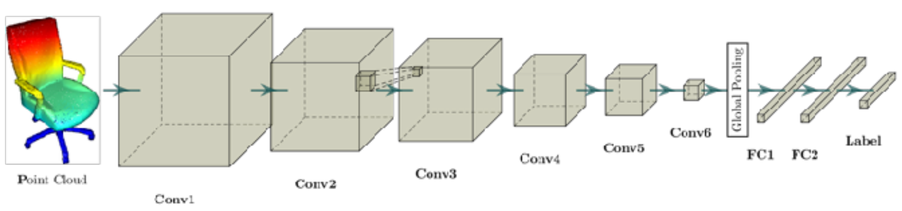
Pontfelhők — Ritkított (sparse) konvolúció

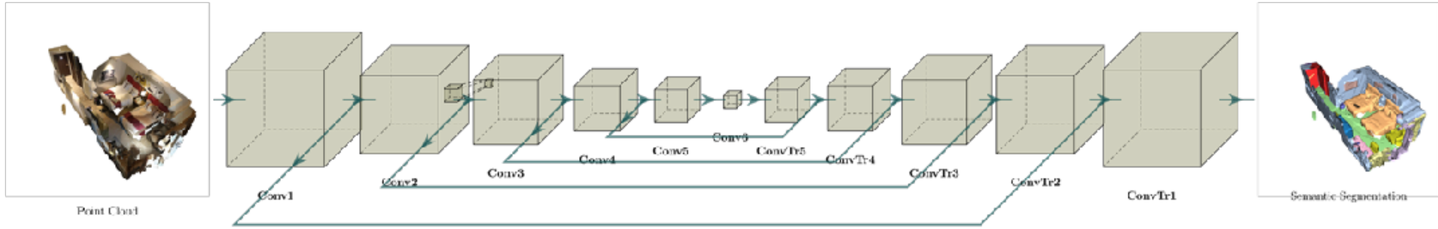


Sűrű konvolúció



Ritkított konvolúció

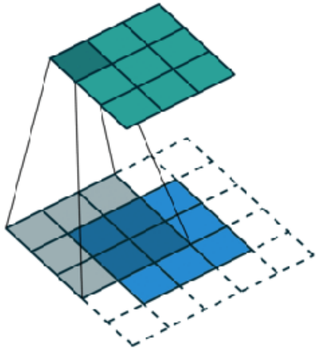




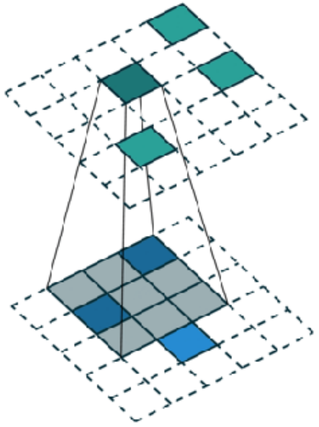
https://nvidia.github.io/MinkowskiEngine/

3D Mélytanulás

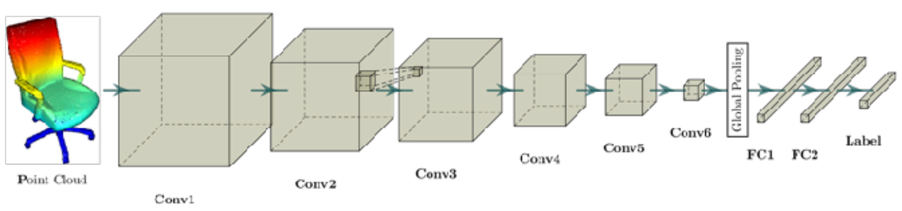
Pontfelhők — Ritkított (sparse) konvolúció

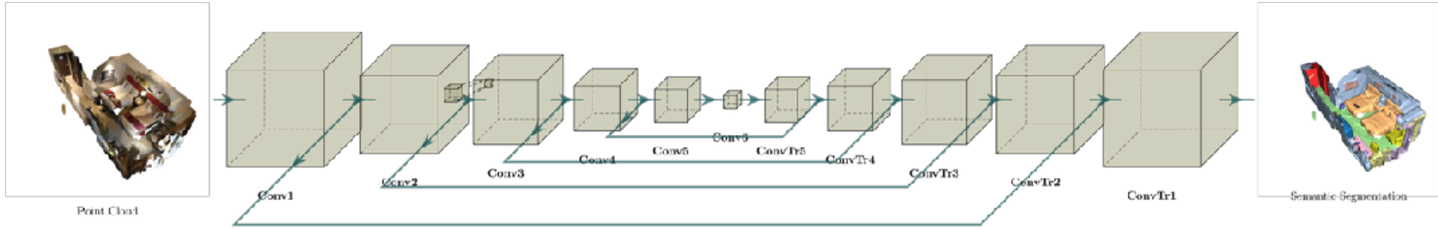


Sűrű konvolúció



Ritkított konvolúció



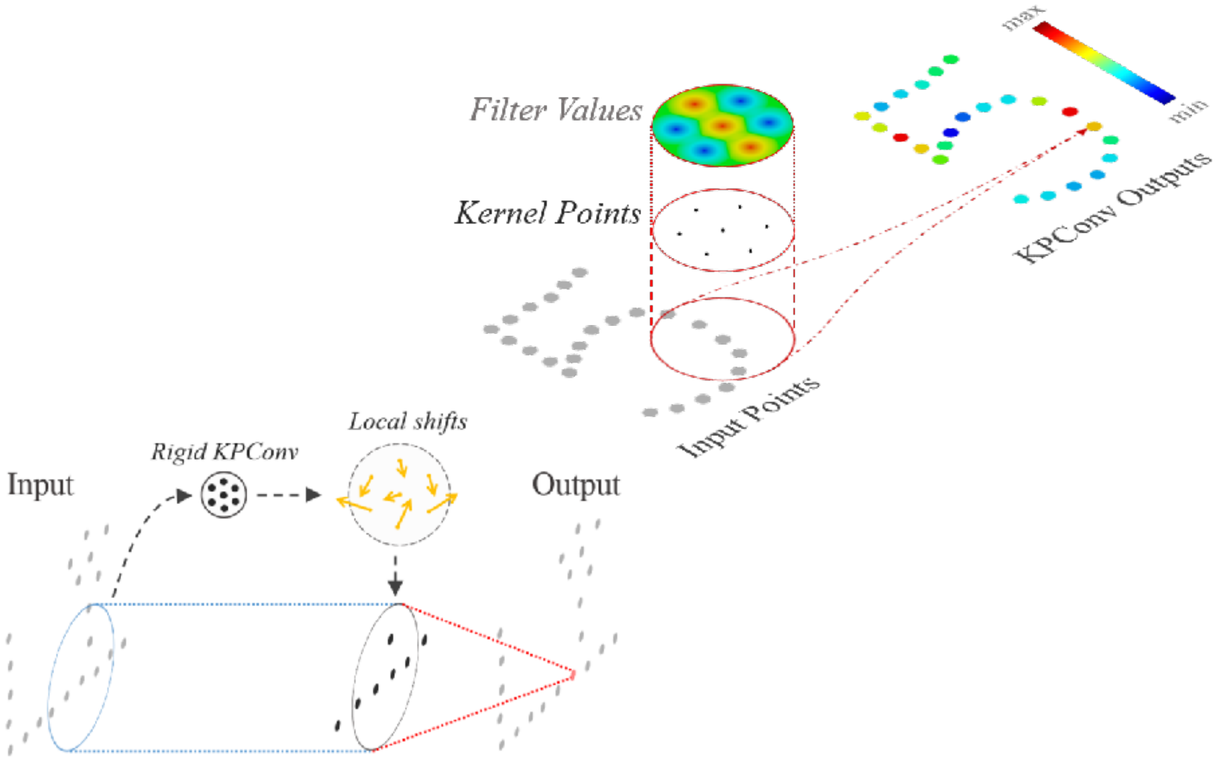


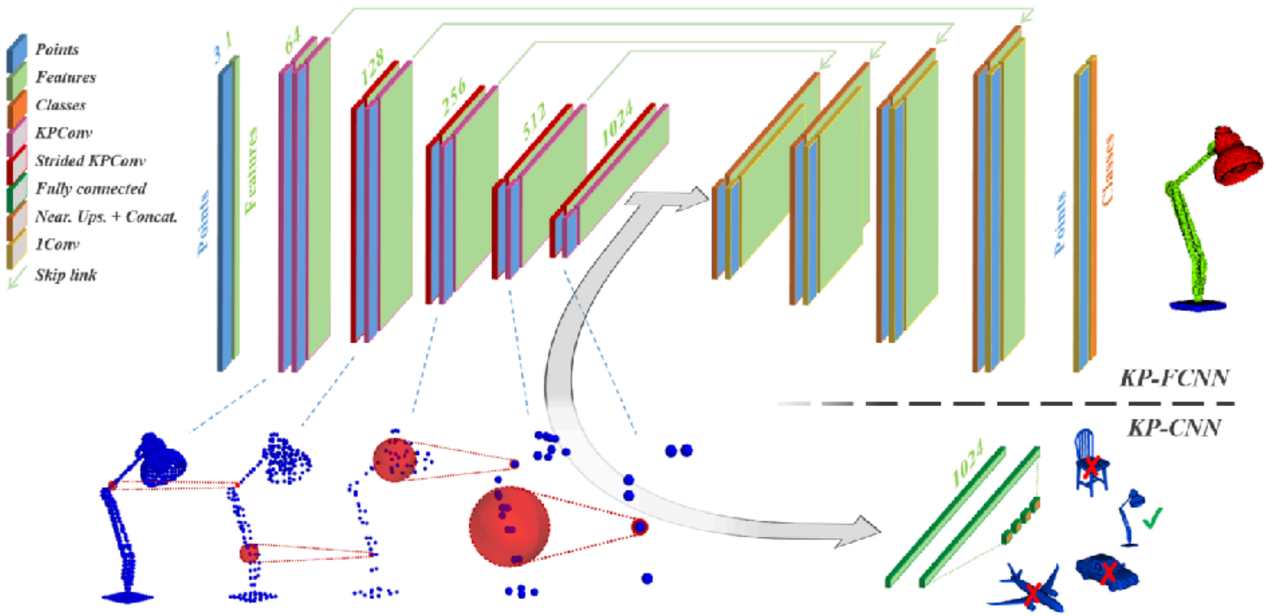
https://nvidia.github.io/MinkowskiEngine/

3D Mélytanulás

Pontfelhők — KPConv

KPConv: Flexible and Deformable Convolution for Point Clouds



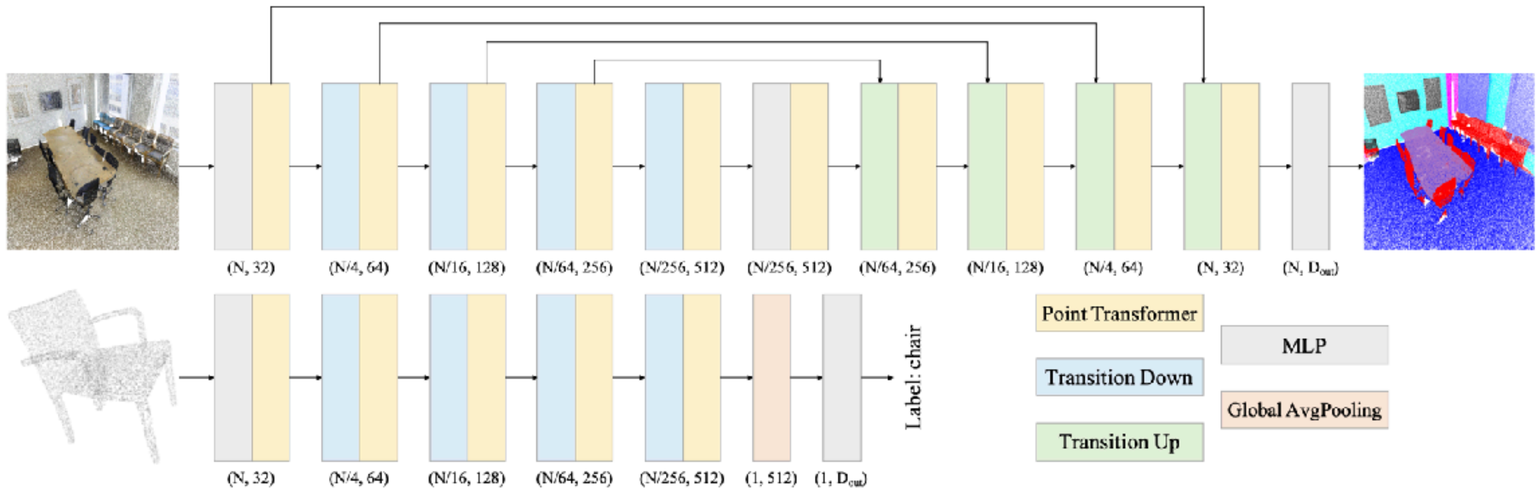


3D Mélytanulás

Pontfelhők — Point Transformer

Point Transformer

Hengshuang Zhao[1] Li Jiang[2] Jiaya Jia[2] Philip Torr[1] Vladlen Koltun[3]



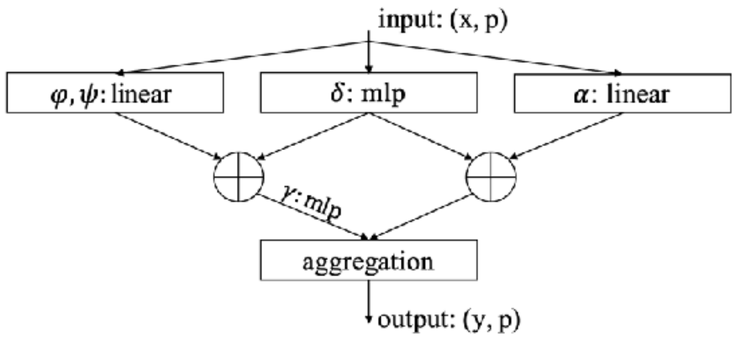
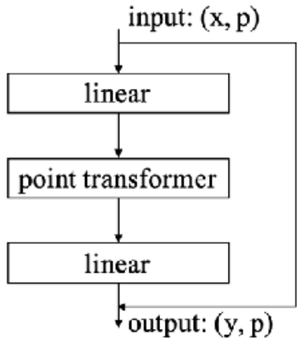
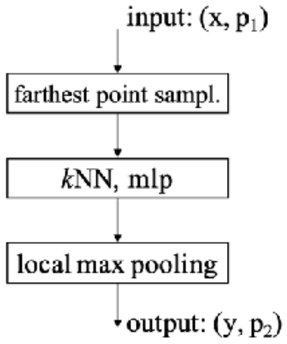


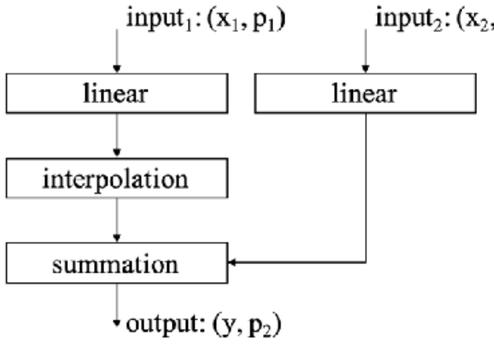
Figure 2. Point transformer layer.



(a) point transformer block



(b) transition down



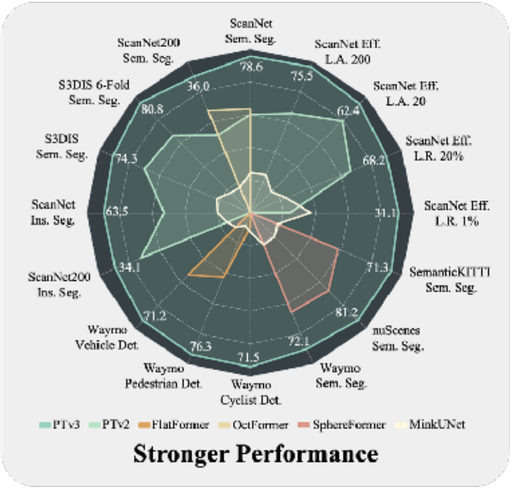
(c) transition up

3D Mélytanulás

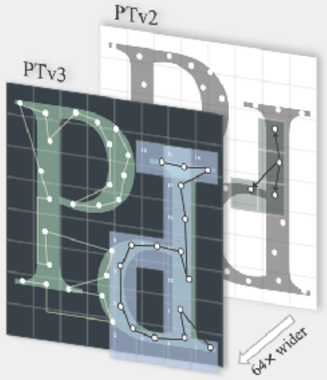
Pontfelhők — Point Transformer V3

Point Transformer V3: Simpler, Faster, Stronger

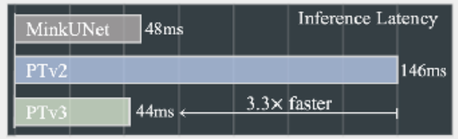
Xiaoyang Wu[1,2] Li Jiang[3] Peng-Shuai Wang[4]
Zhijian Liu[5] Xihui Liu[1] Yu Qiao[2] Wanli Ouyang[2] Tong He[2*] Hengshuang Zhao[1*]



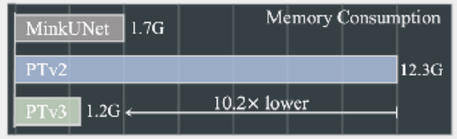
Stronger Performance



Wider Receptive Field



Faster Speed



Lower Memory Consumption

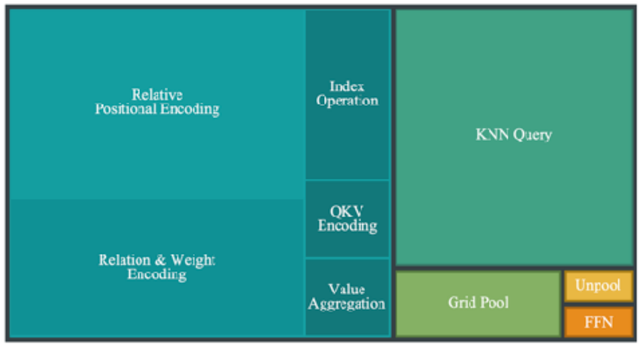
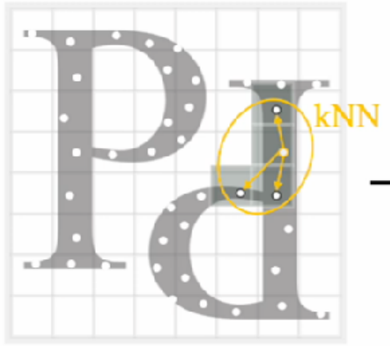
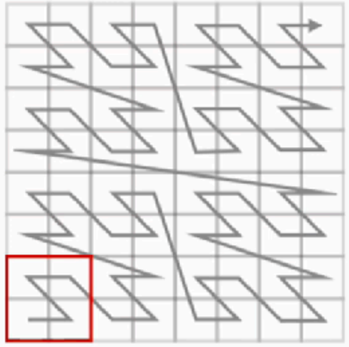


Figure 2. **Latency treemap of each components of PTv2.** We benchmark and visualize the proportion of the forward time of each component of PTv2. KNN Query and RPE occupy a total of 54% of forward time.

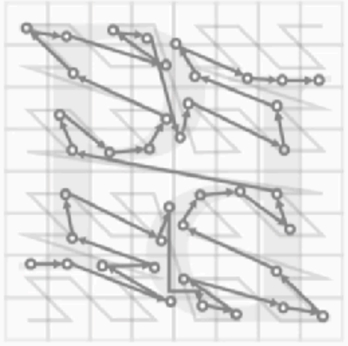


PTv2

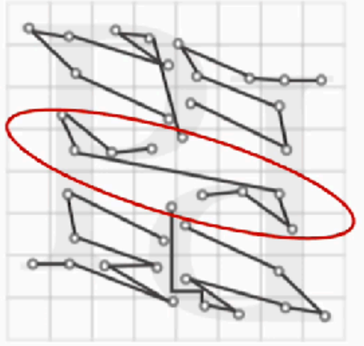
PTv3



Space-filling Curve



Ordered Point Cloud
structured 1D format



Patch Partition
preserving spatial
neighbor relationships

3D Mélytanulás

Pontfelhők — Pointcept könyvtár

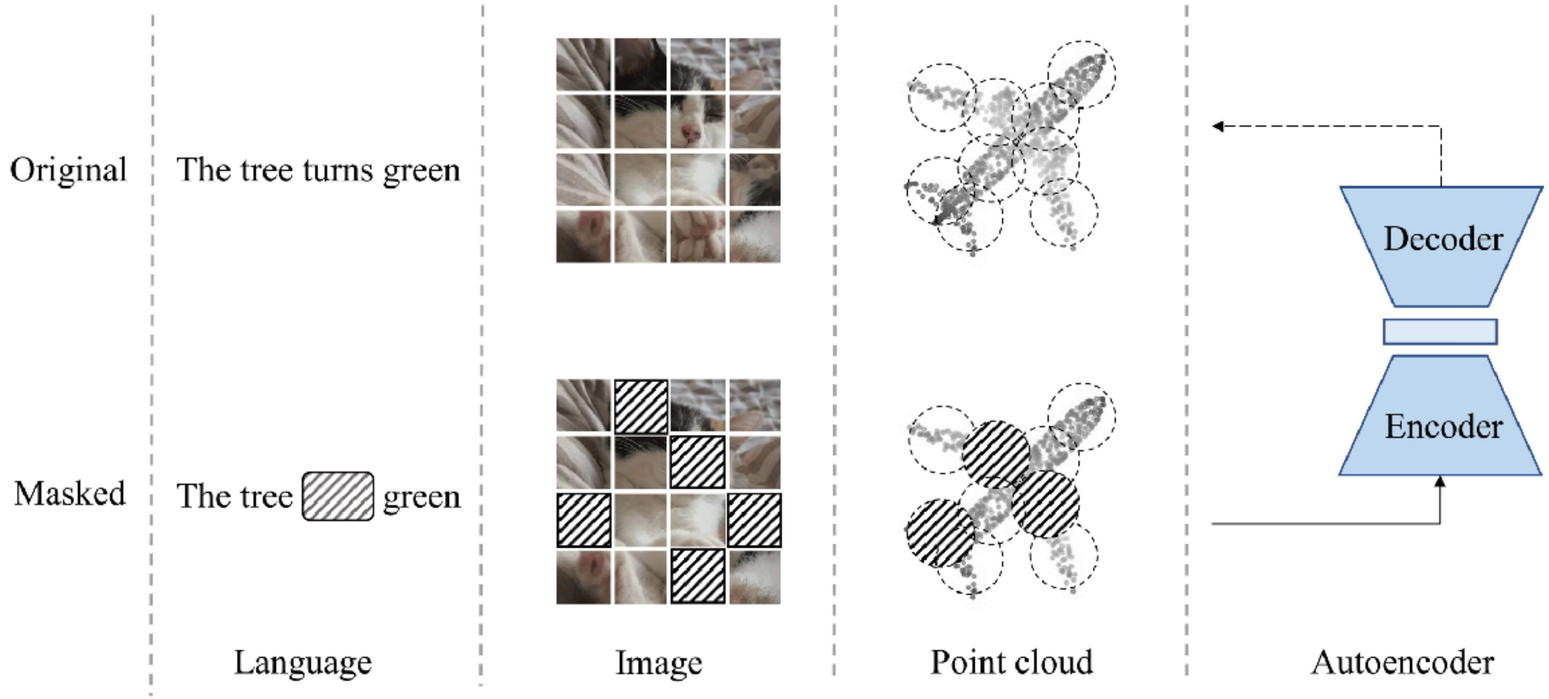
POINTCEPT

Point Cloud Per**cept**ion Codebase

https://github.com/Pointcept/Pointcept

3D Mélytanulás

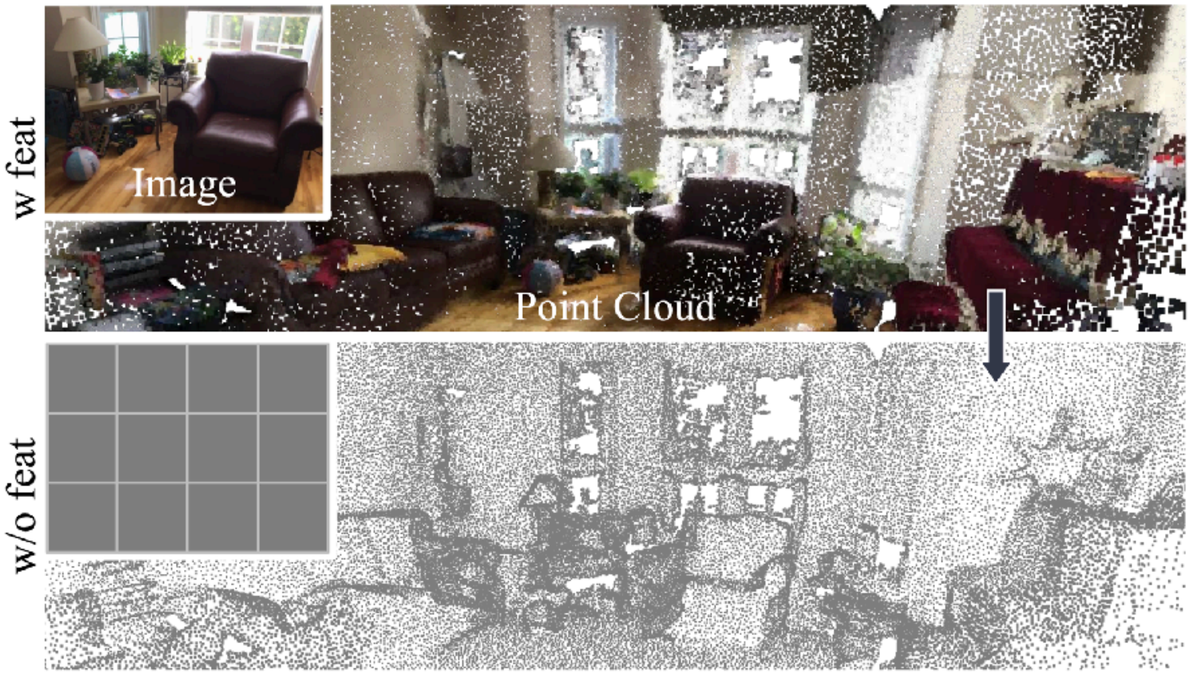
Pontfelhők — Önfelügyelt tanítás

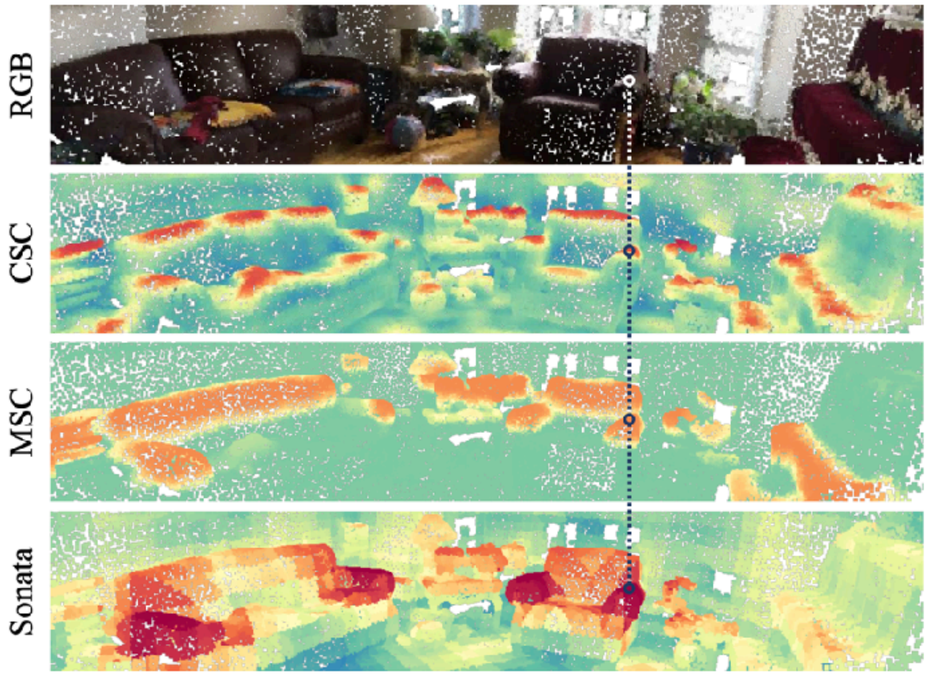


Önfelügyelt tanítás pontfelhőkön?

3D Mélytanulás

Pontfelhők — Önfelügyelt tanítás

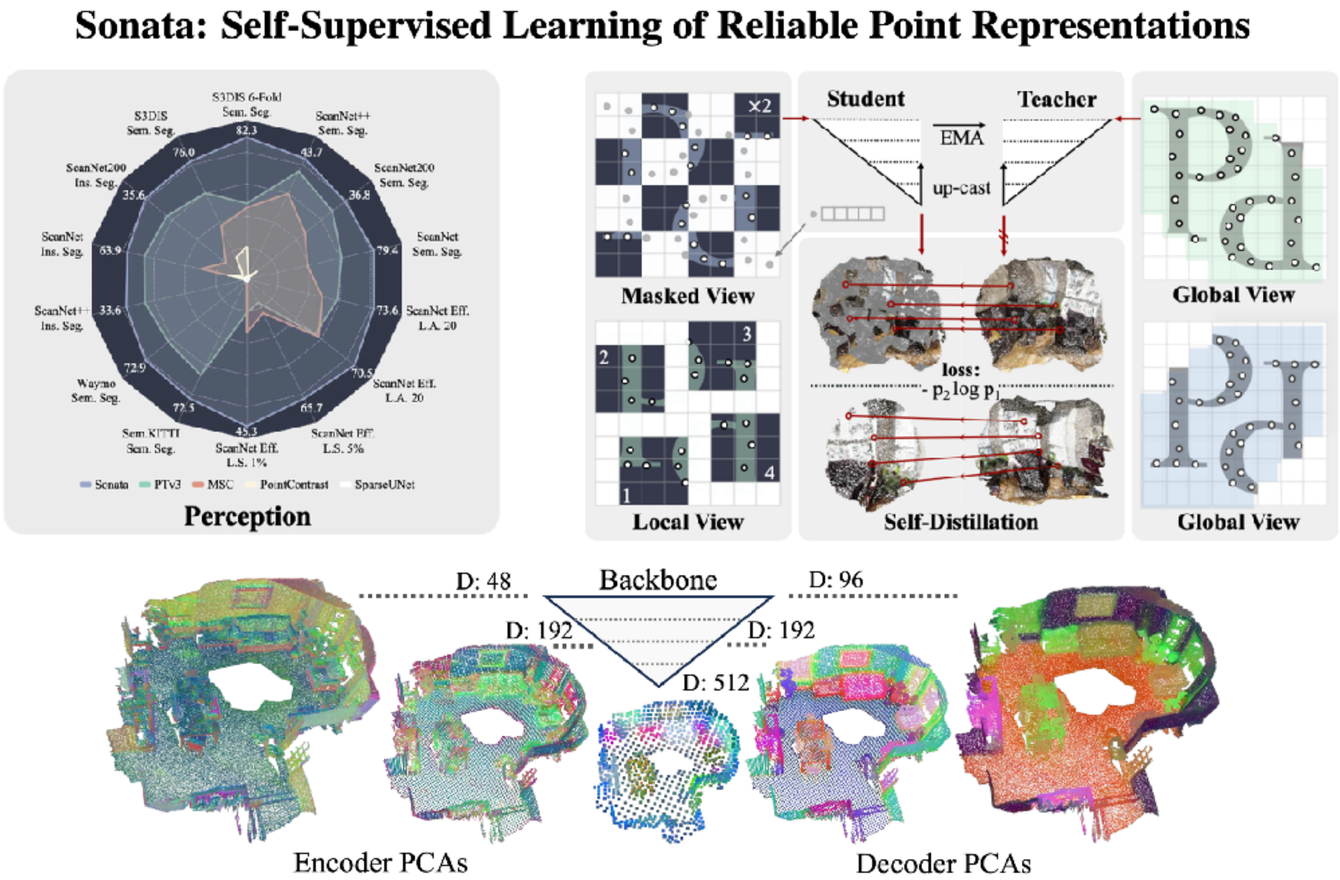




<u>Geometriai “shortcut”</u>:
A pontok “elárulják” a tanulni kívánt geometriai információt a koordinátáik által

3D Mélytanulás

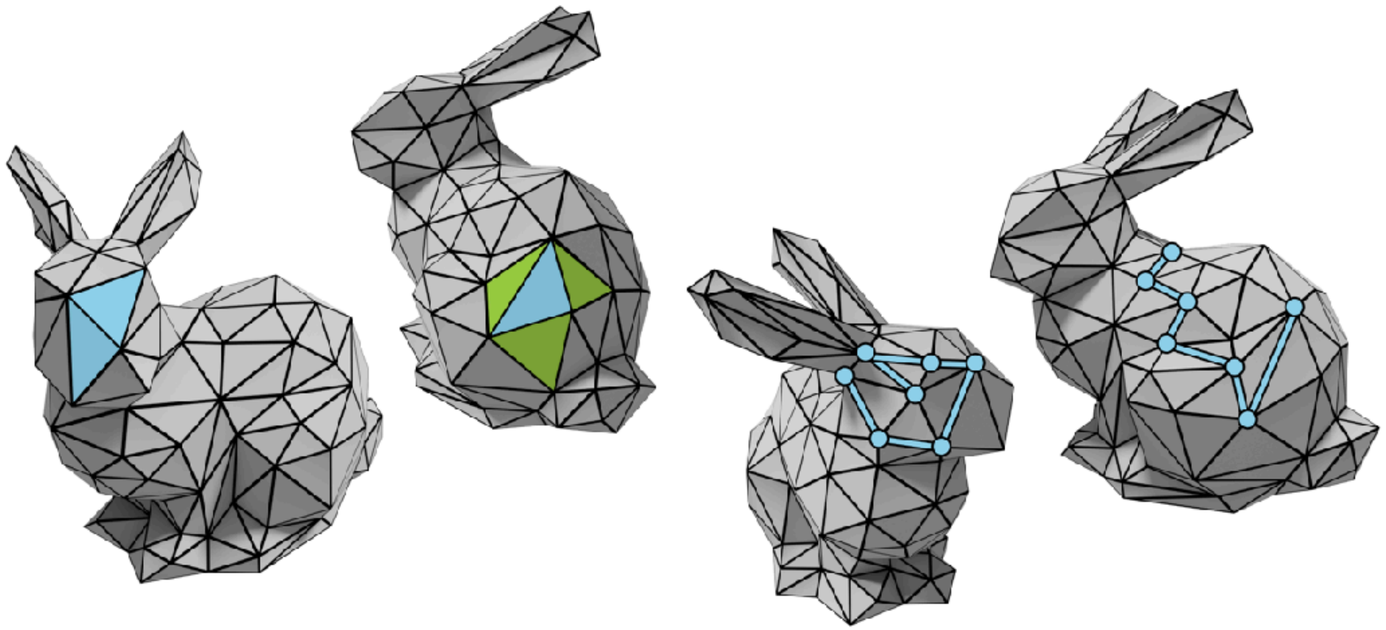
Pontfelhők — Önfelügyelt tanítás



https://xywu.me/sonata/

3D Mélytanulás

Poligonhálók

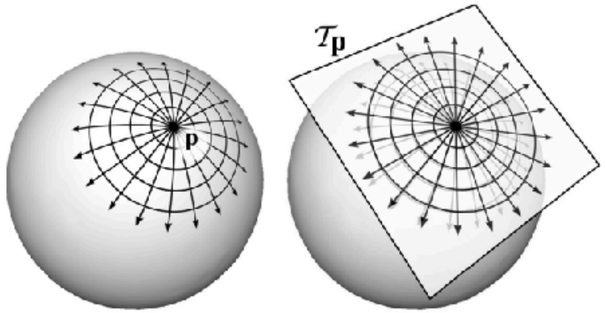


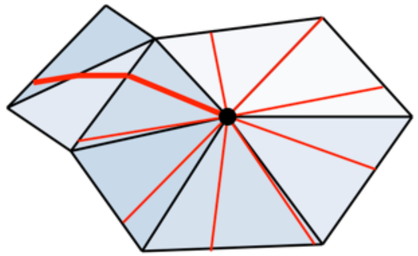
https://anintroductiontodeeplearningonmeshes.github.io/

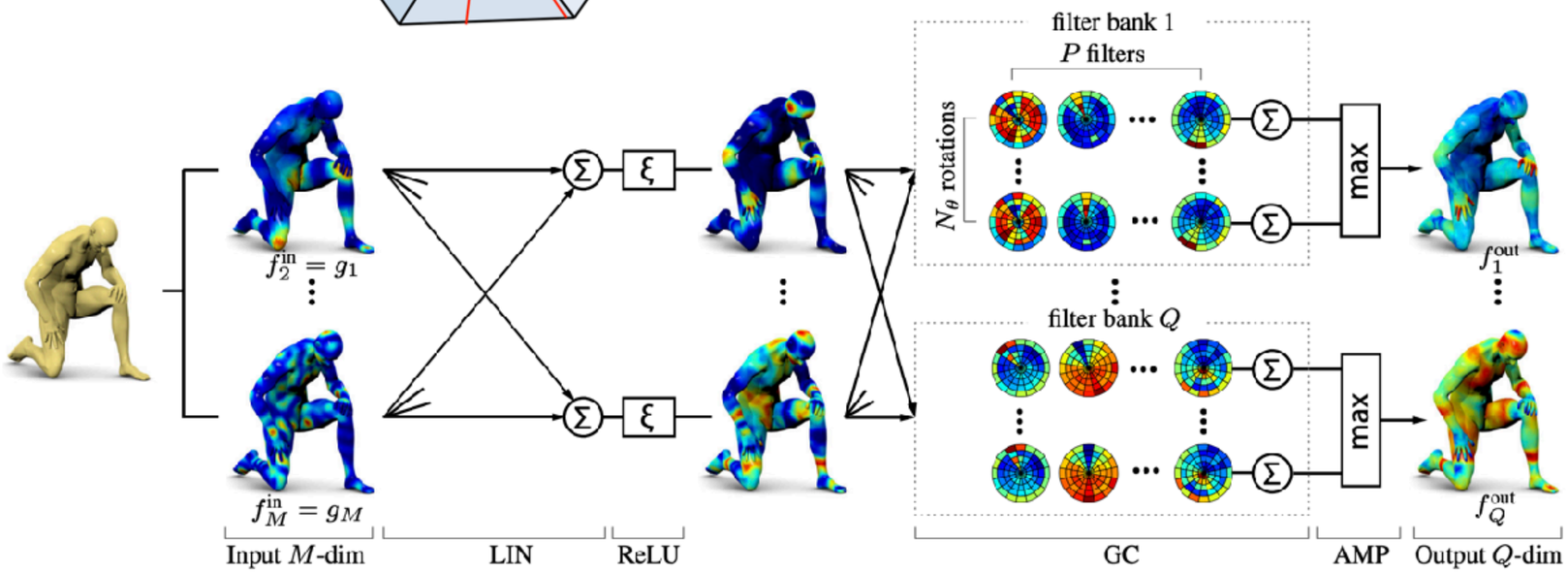
3D Mélytanulás

Poligonhálók — Felületi CNN

Geodesic convolutional neural networks on Riemannian manifolds

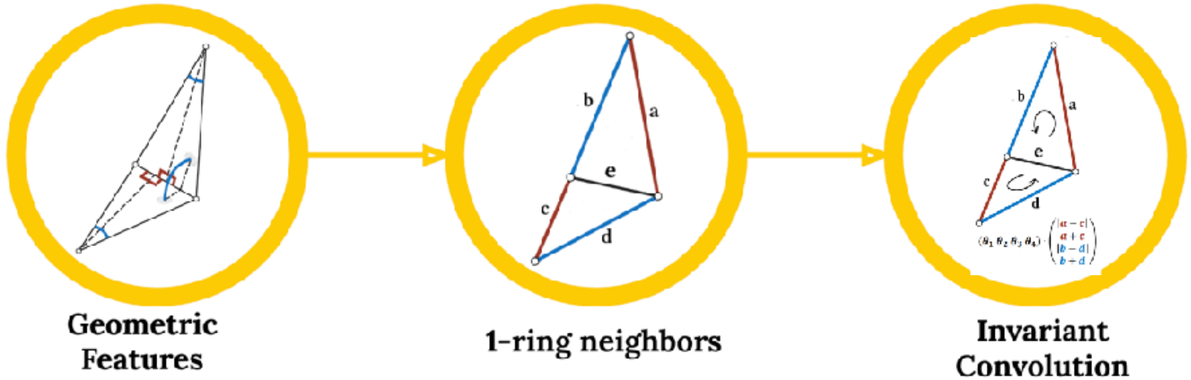


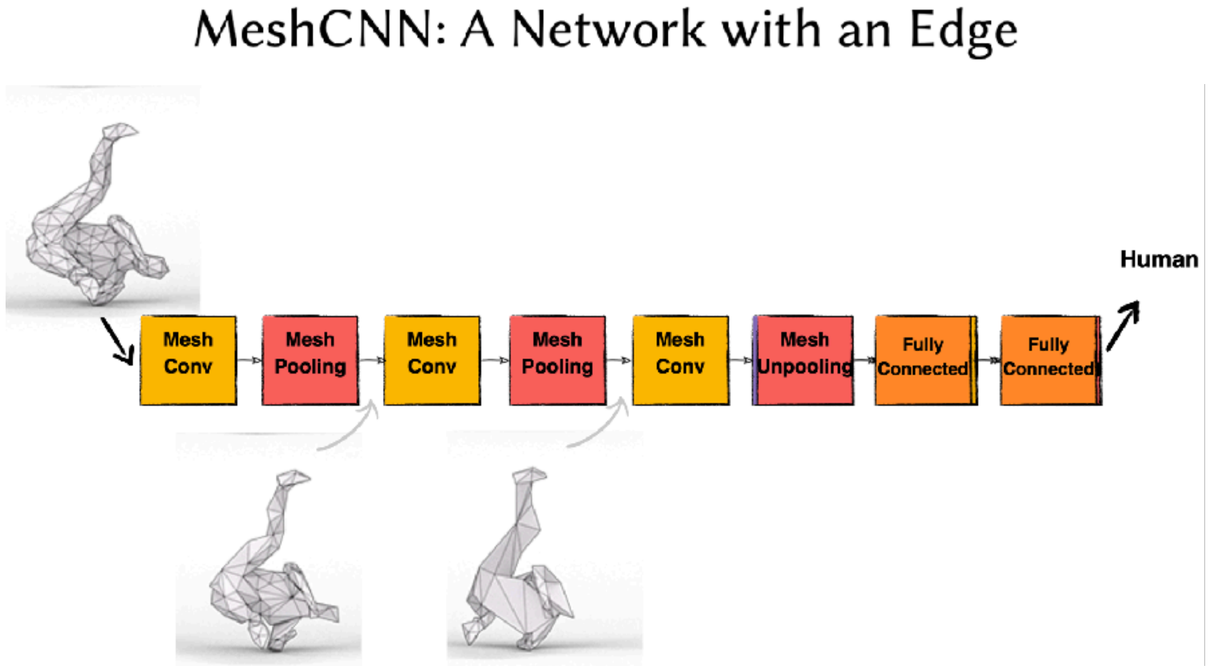


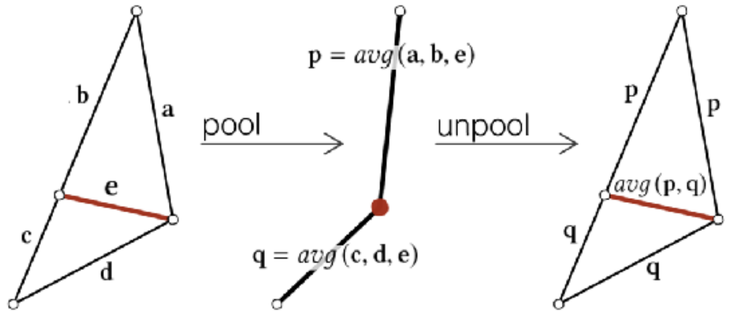


3D Mélytanulás

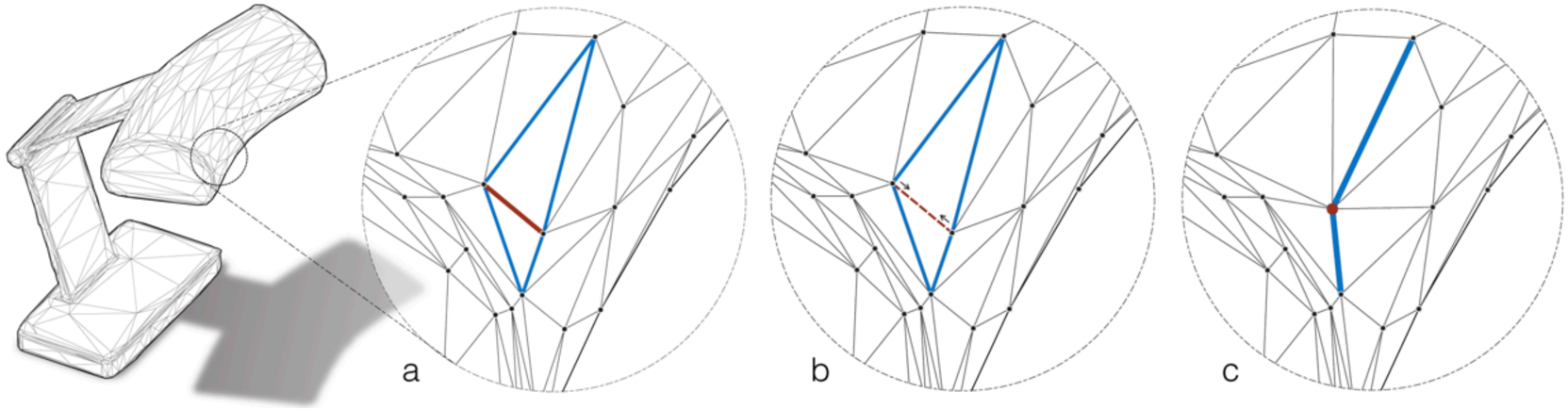
Poligonhálók — MeshCNN







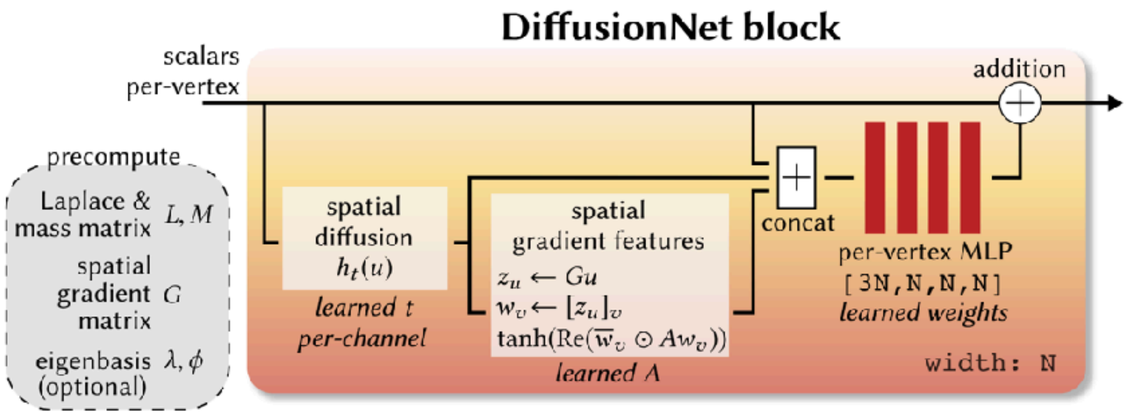
Mesh Pooling & Unpooling

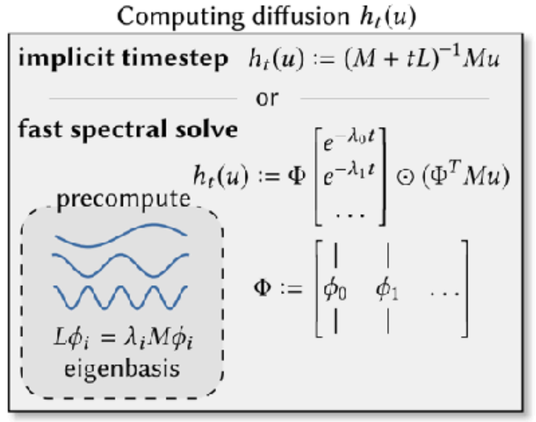


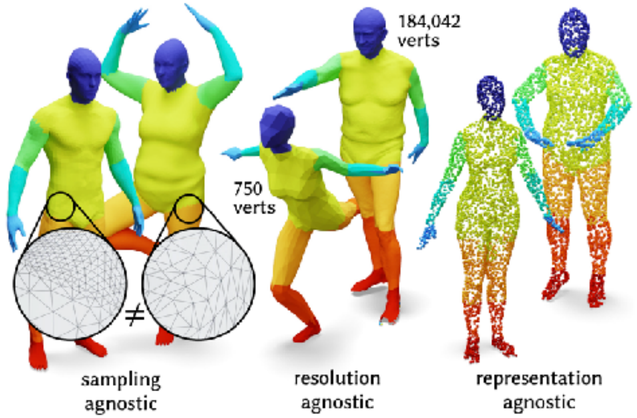
3D Mélytanulás

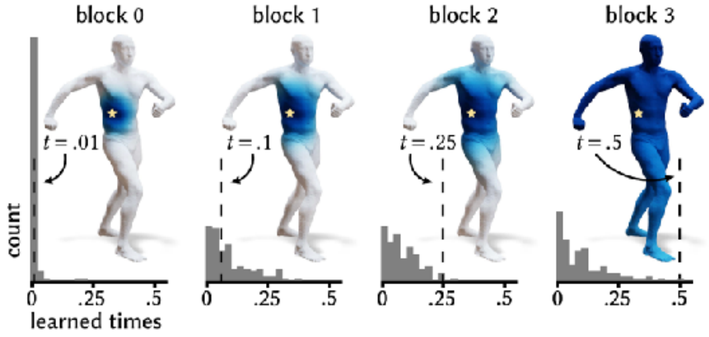
Poligonhálók — DiffusionNet

DiffusionNet: Discretization Agnostic Learning on Surfaces





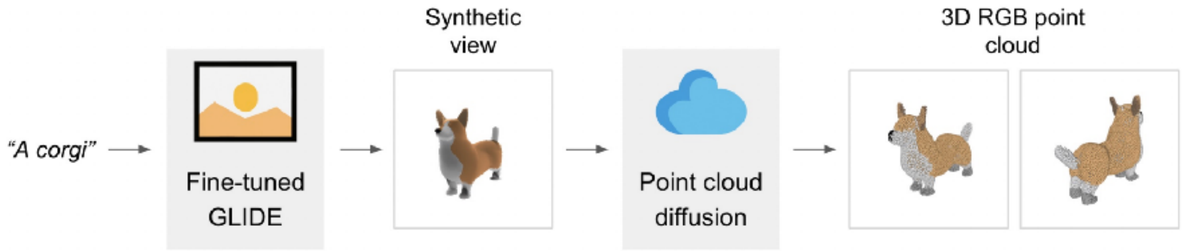


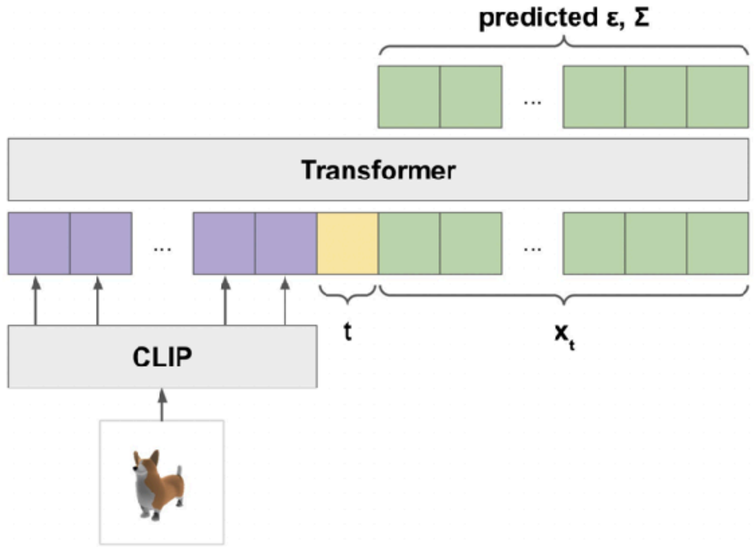


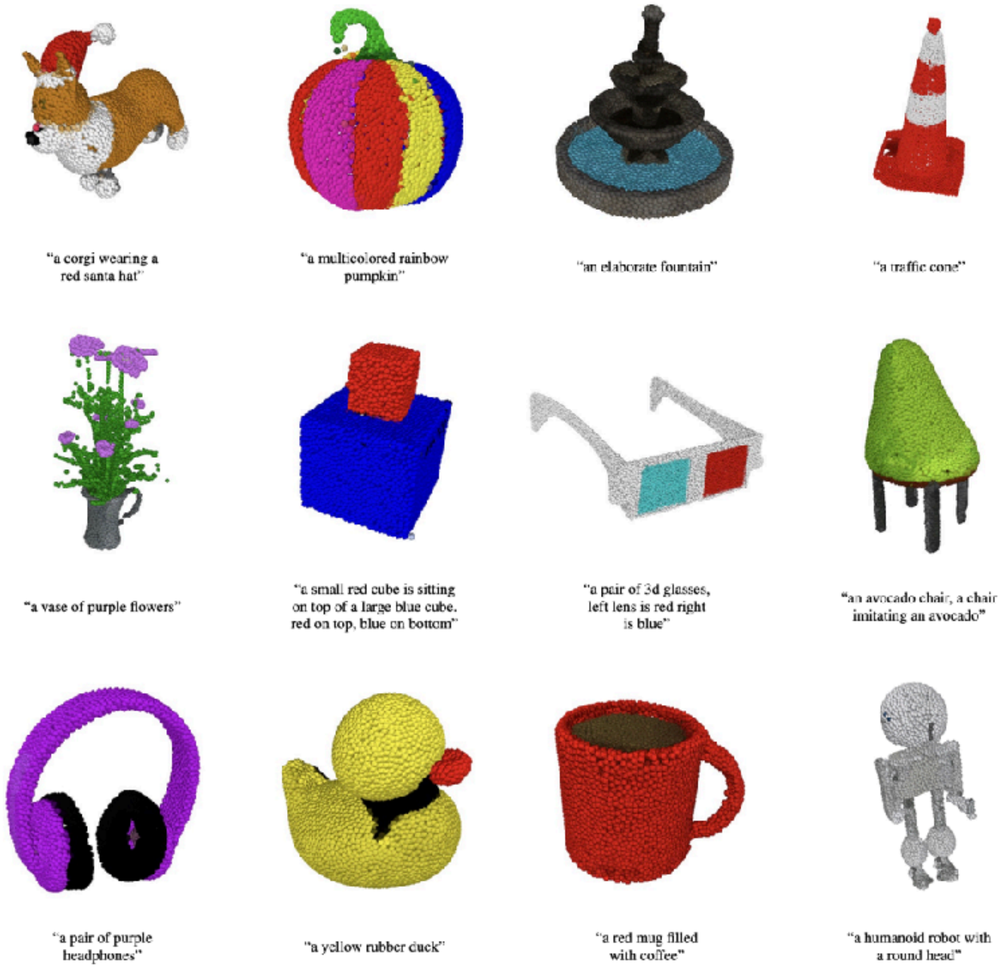
Method	Accuracy
GWCNN [Ezuz et al. 2017]	90.3%
MeshCNN† [Hanocka et al. 2019]	91.0%
HSN† [Wiersma et al. 2020]	96.1%
MeshWalker† [Lahav and Tal 2020]	97.1%
PD-MeshNet† [Milano et al. 2020]	99.1%
HodgeNet† [Smirnov and Solomon 2021]	94.7%
FC† [Mitchel et al. 2021]	99.2%
DiffusionNet - xyz†	99.4%
DiffusionNet - xyz	99.0%
DiffusionNet - hks†	99.5%
DiffusionNet - hks	**99.7%**

3D Generálás

Pontfelhő diffúzió

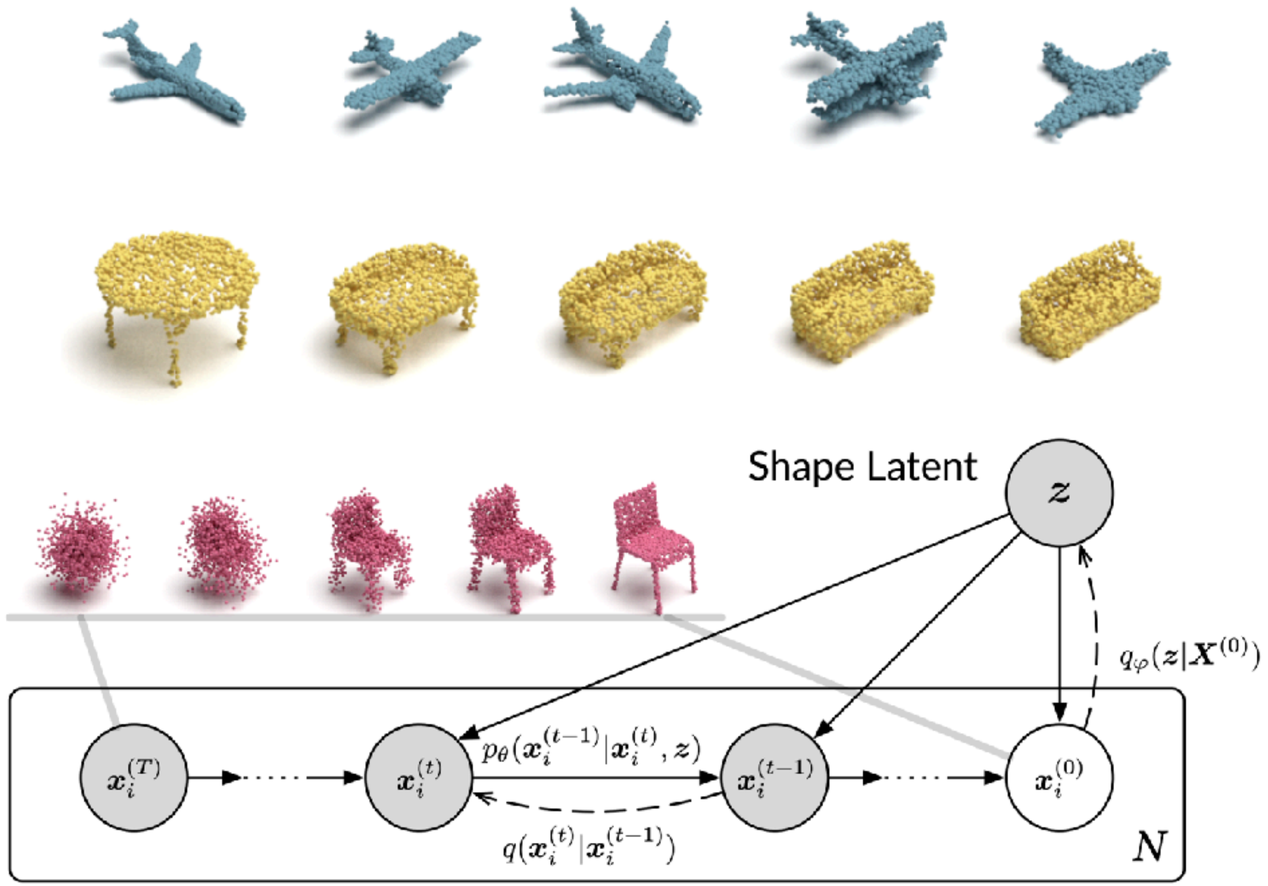






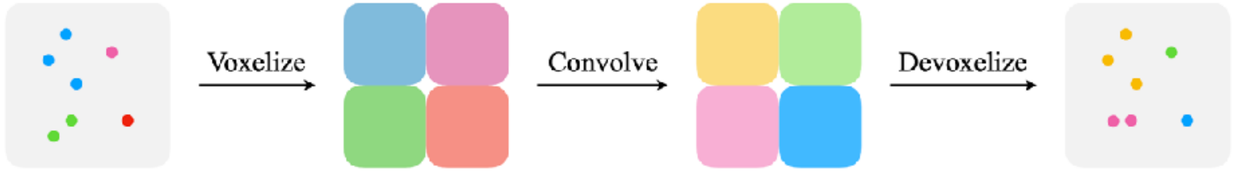
3D Generálás

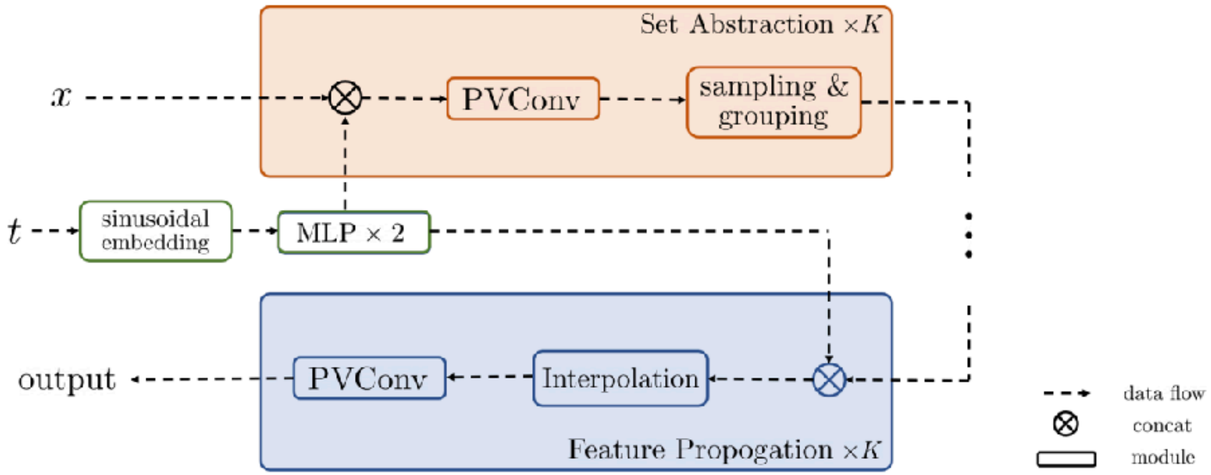
Ponfelhő diffúzió

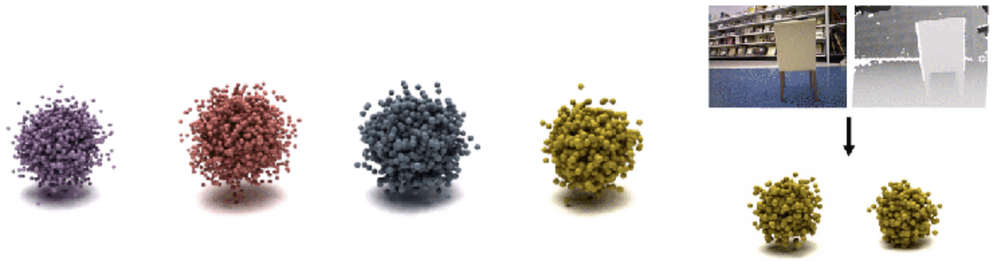


3D Generálás

Pont-voxel konvolúció

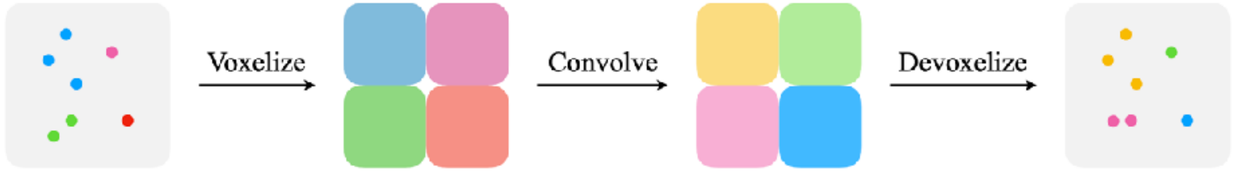


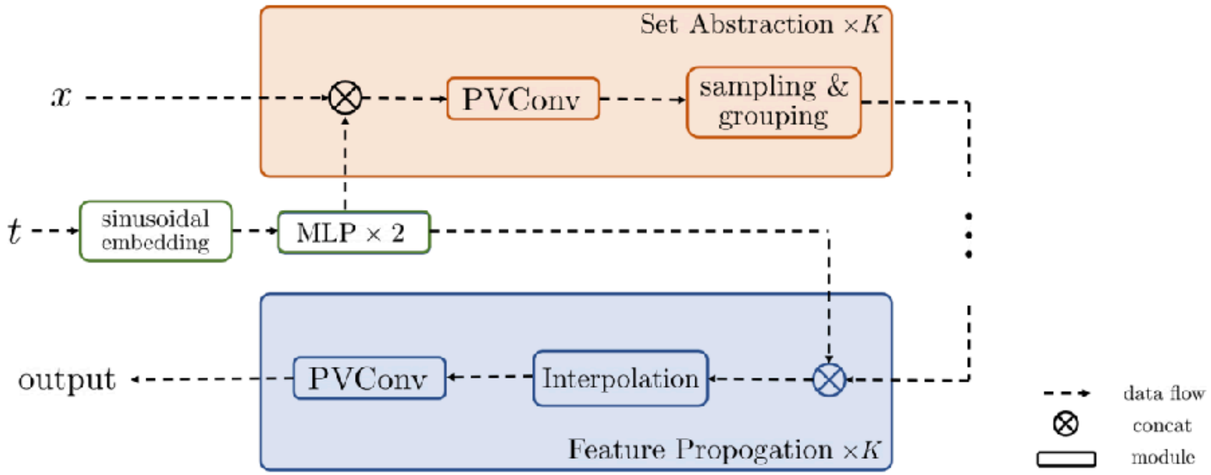


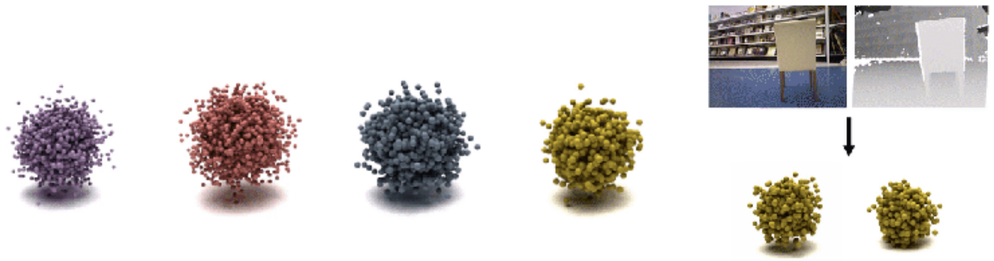


3D Generálás

Pont-voxel konvolúció

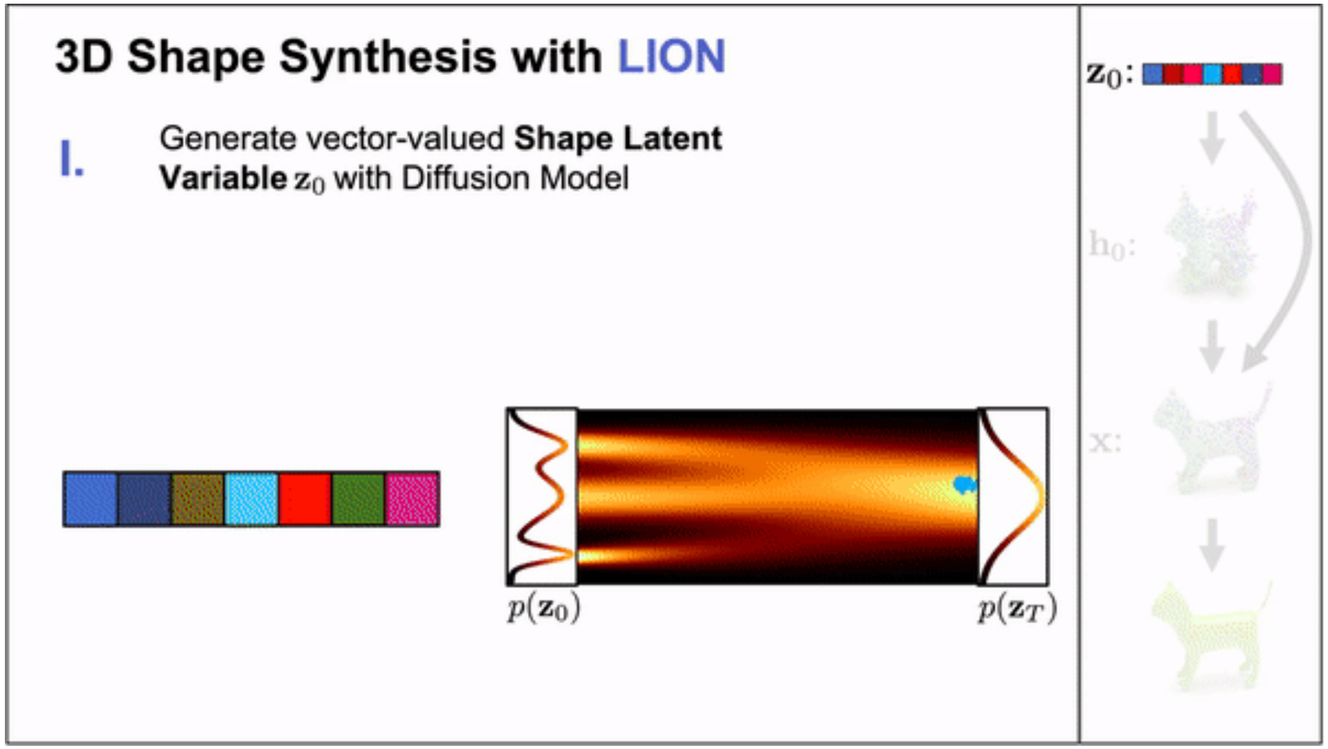






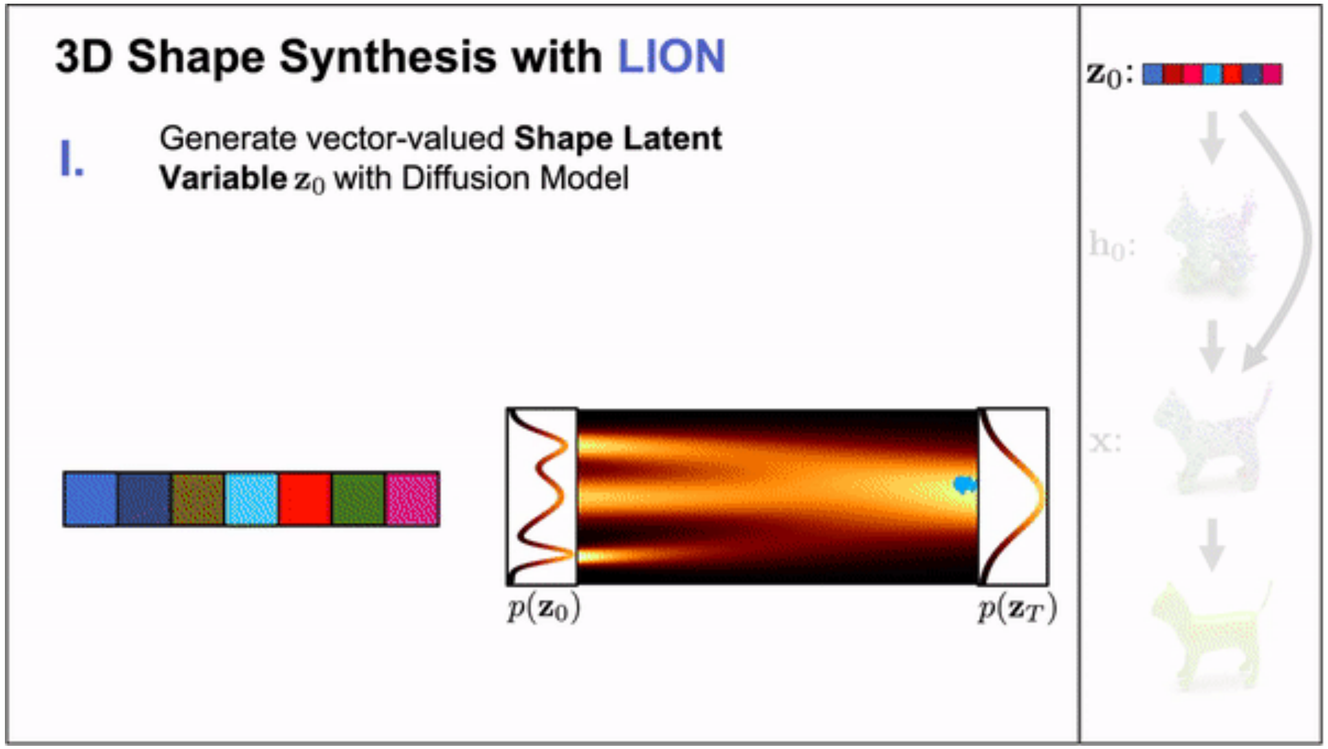
3D Generálás

Látens pontfelhő diffúzió



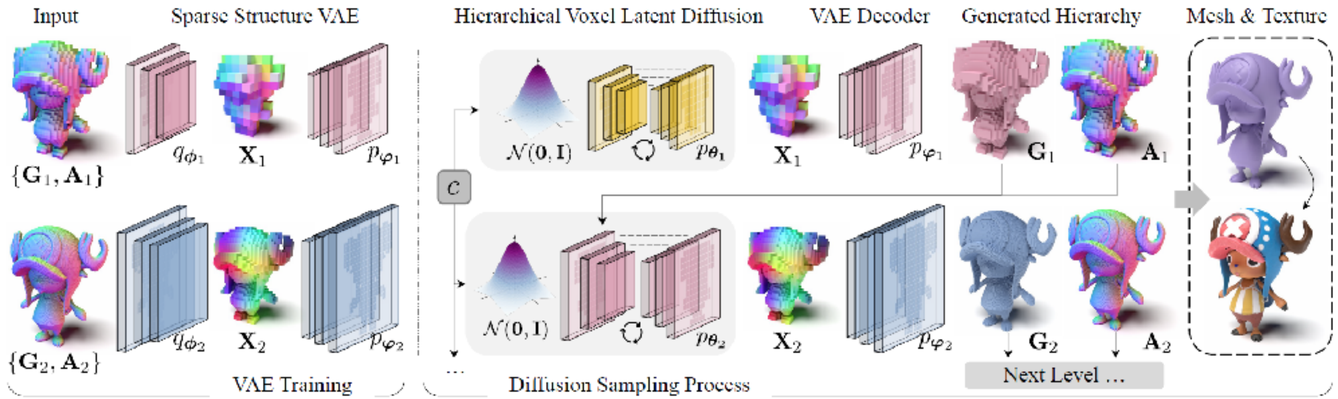
3D Generálás

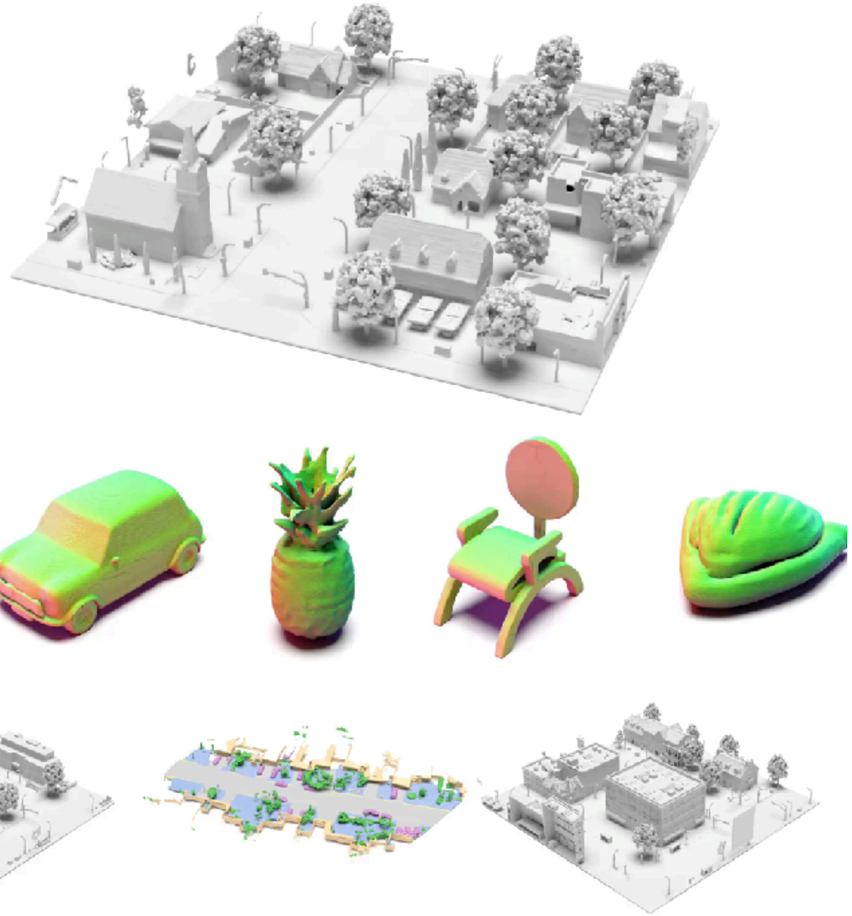
Látens pontfelhő diffúzió



3D Generálás

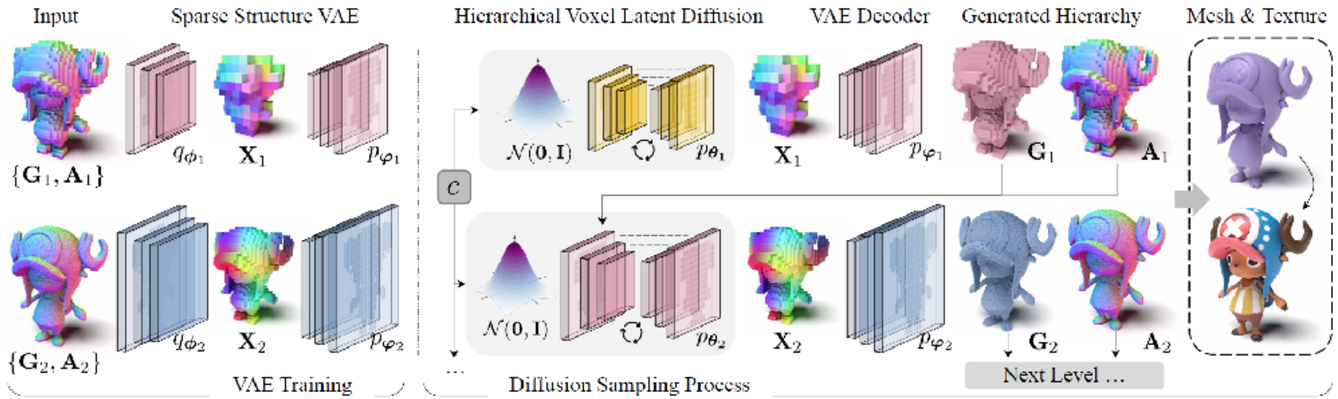
Látens voxel diffúzió

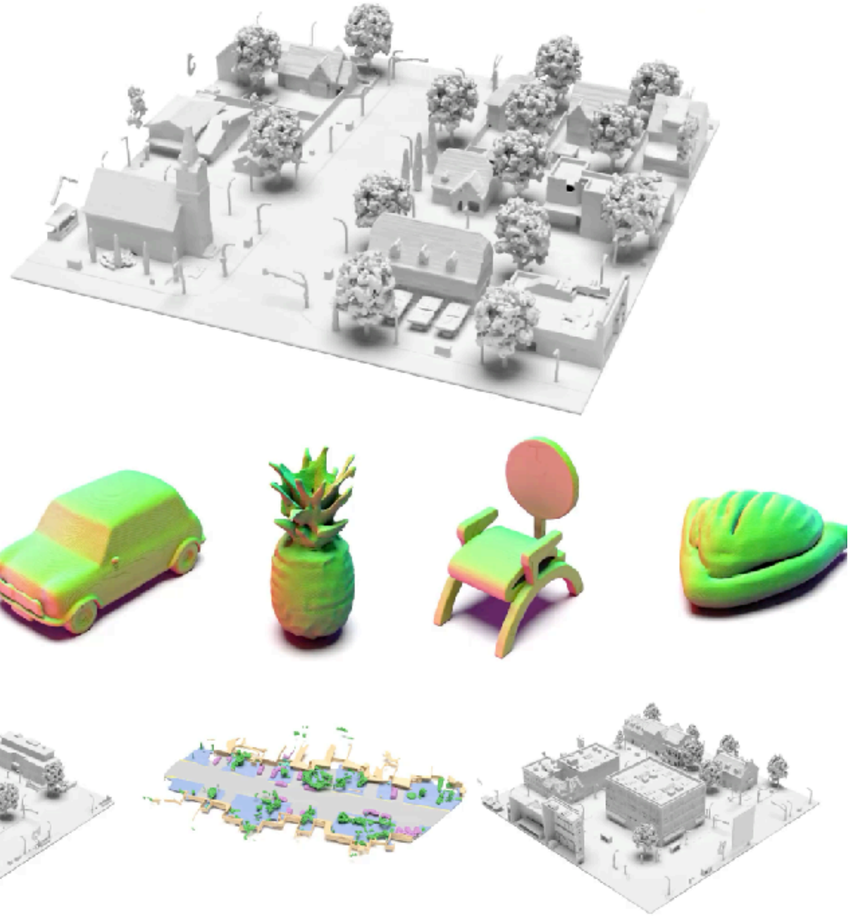




3D Generálás

Látens voxel diffúzió





3D Generálás

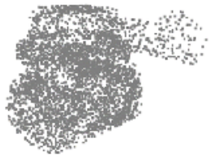
Háló generálás — Autoregresszió

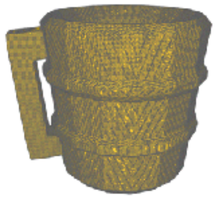
MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers

Yawar Siddiqui[1] Antonio Alliegro[2] Alexey Artemov[1]
Tatiana Tommasi[2] Daniele Sirigatti[3] Vladislav Rosov[3] Angela Dai[1] Matthias Nießner[1]
Technical University of Munich[1] Politecnico di Torino[2] AUDI AG[3]

MESHANYTHING: ARTIST-CREATED MESH GENERATION WITH AUTOREGRESSIVE TRANSFORMERS

Yiwen Chen[1,2]*, Tong He[2†], Di Huang[2], Weicai Ye[2], Sijin Chen[3], Jiaxiang Tang[4]
Xin Chen[5], Zhongang Cai[6], Lei Yang[6], Gang Yu[7], Guosheng Lin[1†], Chi Zhang[8†]
[1]S-Lab, Nanyang Technological University [2]Shanghai AI Lab
[3]Fudan University [4]Peking University [5]University of Chinese Academy of Sciences
[6]SenseTime Research [7]Stepfun [8]Westlake University
https://buaacyw.github.io/mesh-anything/


Point Cloud


Dense Mesh


NeRF


3D GS

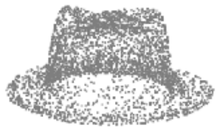

Image

A commode

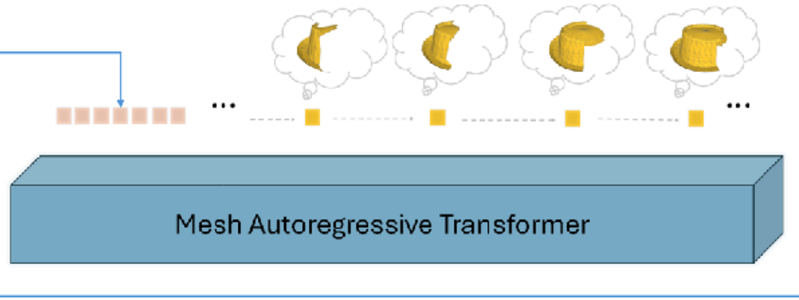
Text

- 3D reconstruction
- Text-to-3D
- Lidar Scanning

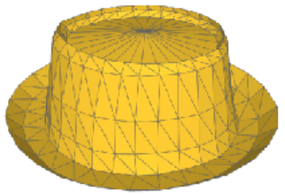
...


Point Cloud


Feature



Decode


Mesh

3D Generálás

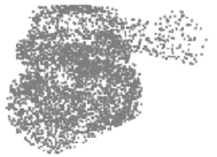
Háló generálás — Autoregresszió

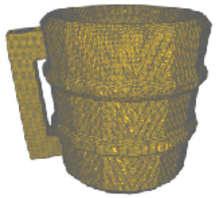
MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers

Yawar Siddiqui[1] Antonio Alliegro[2] Alexey Artemov[1]
Tatiana Tommasi[2] Daniele Sirigatti[3] Vladislav Rosov[3] Angela Dai[1] Matthias Nießner[1]
Technical University of Munich[1] Politecnico di Torino[2] AUDI AG[3]

MESHANYTHING: ARTIST-CREATED MESH GENERATION WITH AUTOREGRESSIVE TRANSFORMERS

Yiwen Chen[1,2]*, Tong He[2†], Di Huang[2], Weicai Ye[2], Sijin Chen[3], Jiaxiang Tang[4]
Xin Chen[5], Zhongang Cai[6], Lei Yang[6], Gang Yu[7], Guosheng Lin[1†], Chi Zhang[8†]
[1]S-Lab, Nanyang Technological University [2]Shanghai AI Lab
[3]Fudan University [4]Peking University [5]University of Chinese Academy of Sciences
[6]SenseTime Research [7]Stepfun [8]Westlake University
https://buaacyw.github.io/mesh-anything/


Point Cloud


Dense Mesh


NeRF


3D GS

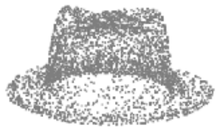

Image

A commode

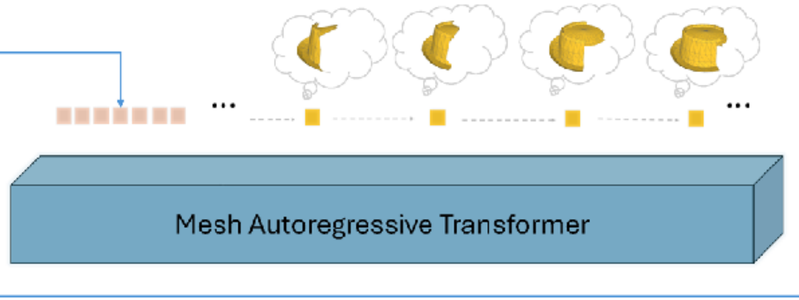
Text

- 3D reconstruction
- Text-to-3D
- Lidar Scanning

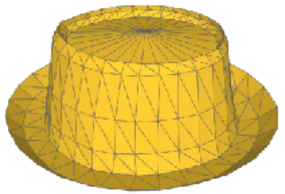
...


Point Cloud


Feature



Decode


Mesh

Következő előadás:

Neurális implicit reprezentációk