3D shape understanding

682: Neural Networks: A Modern Introduction

Subhransu Maji

April 21, 2026

College of
INFORMATION AND
COMPUTER SCIENCES

UMASS
AMHERST

Agenda

3D representations

3D recognition architectures

- Muti-view methods
- Voxel-based methods
- Point-based methods

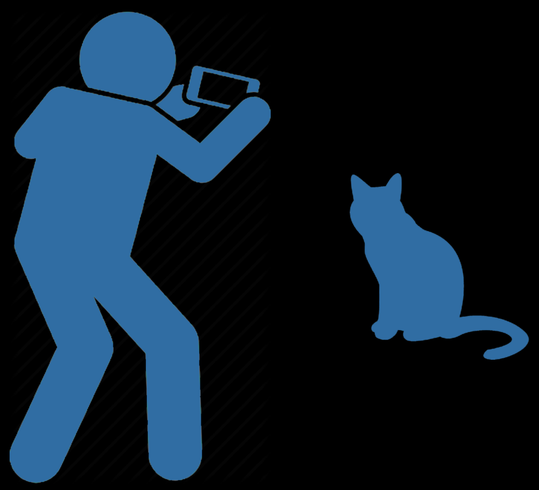
Recent trends

- Image to 3D

Slides credits: Hang Su & Hao Su

3D World Through 2D Lenses

Traditionally, visual recognition algorithms rely primarily on **2D input signals**, even though the targets are almost always **3D entities**.



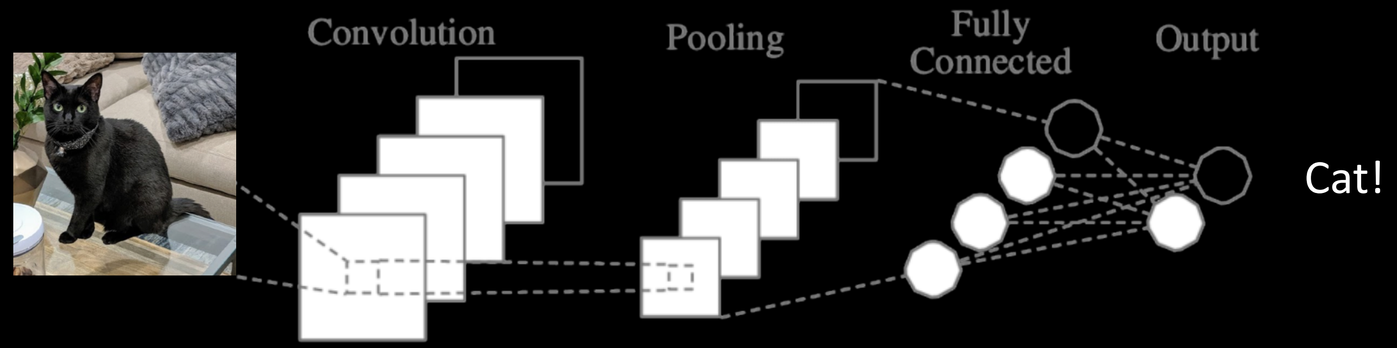
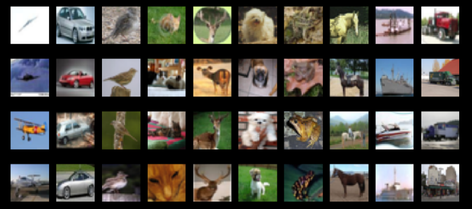


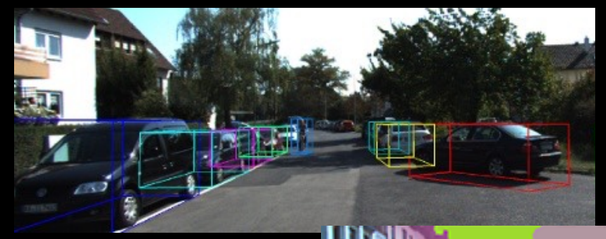
Image Classification

3D World Through 2D Lenses

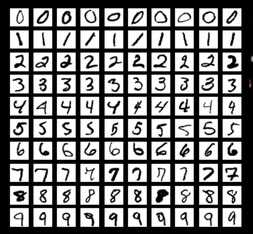
Traditionally, visual recognition algorithms rely primarily on **2D input signals**, even though the targets are almost always **3D entities**.



CIFAR-10



KITTI



MNIST

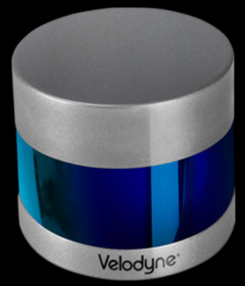
PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

COCO
Common Objects in Context

A Growing Interest in 3D Recognition Models

- 3D sensors have become better and more accessible.
- Applications in autonomous driving, robotics, consumer electronics, etc.

HDL-64E ($85k, 2007)



VLP-16 ($8k, 2014)

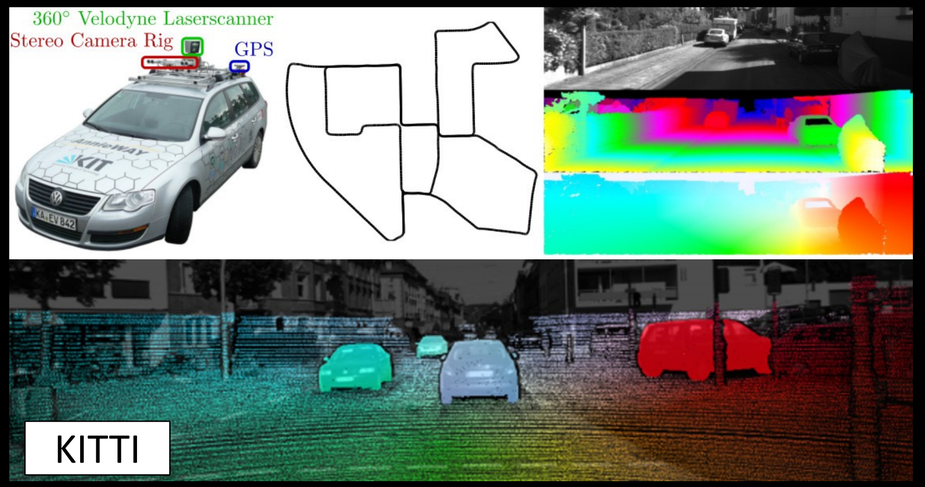
OS-1-16 ($3.5k, 2017)

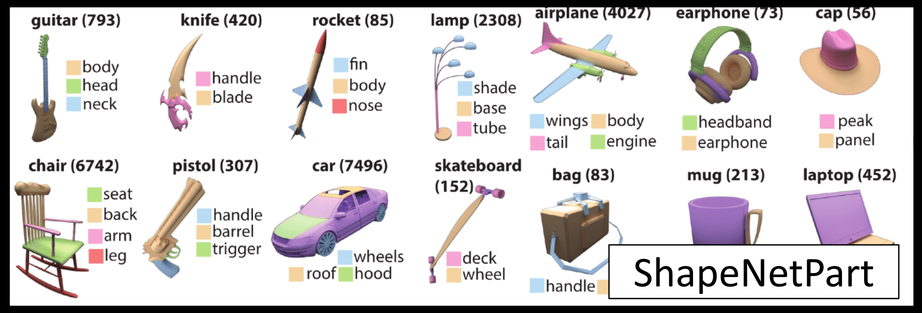
Velabit ($100, 2020)

Livox Horizon ($1k, 2020)

A Growing Interest in 3D Recognition Models

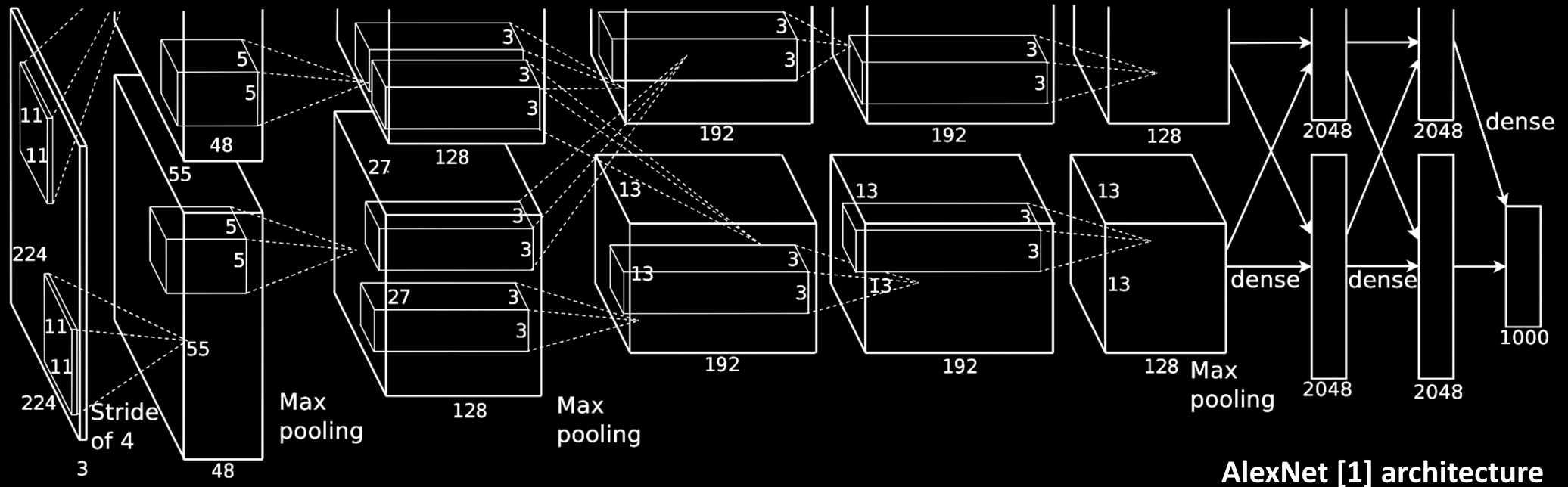
- Large-scale 3D datasets with annotations become available.





Challenges

- Convolutional Neural Networks (CNNs) achieve great success in 2D visual recognition tasks.
- Extending such models to 3D naively requires **dense regular grid structure** .



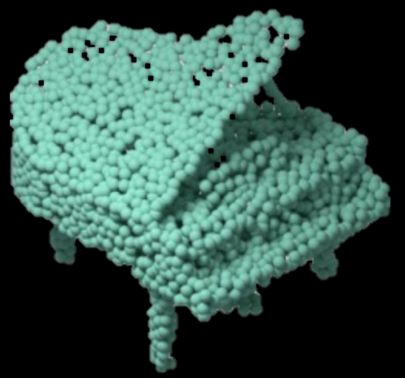
AlexNet [1] architecture

1. A. Krizhevsky, I. Sutskever and G. E. Hinton. Imagenet classification with deep convolutional neural networks. NIPS '12

Challenges

- Common representations of raw 3D data:
  - Polygon meshes (e.g. from CAD models)
  - Point clouds (e.g. from LiDAR sensors)

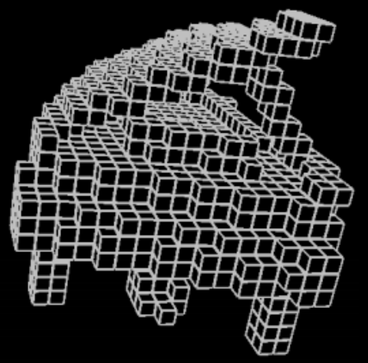
Polygon Meshes



Point Clouds

- Both lack regular grid structures to perform spatial convolution.

- A workaround: converting to voxels (volumetric formats)
  - √ 3D spatial convolutions readily apply
  - × Losing details
  - × Efficiency trade-offs



Voxels

3D shape classification

chair

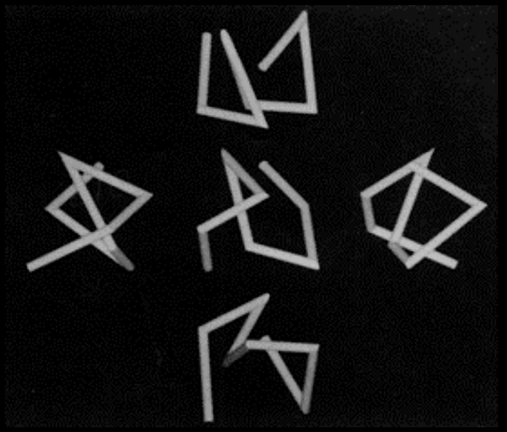
Multi-view CNN

Image-Based 3D Representations

Can the 3D world be effectively represented with its 2D views?

Evidences human brain models objects as “collections of views”

- *Faster* and *more accurate* to recognize when given some views of objects
- Some neurons were found to respond selectively to particular views



[Bülthoff and Edelman, '91]

Some existing 3D descriptors are indeed image-based

- E.g. Light Field Descriptor
- Also work based on Fisher vectors, Gabor filters, SIFT features, etc.

[Tarr, '95]

Image-Based 3D Representations

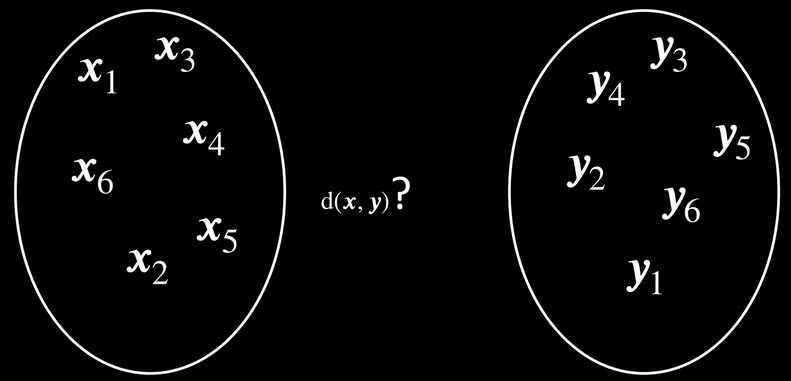
Can we use 2D CNNs for 3D recognition?

Benefits:

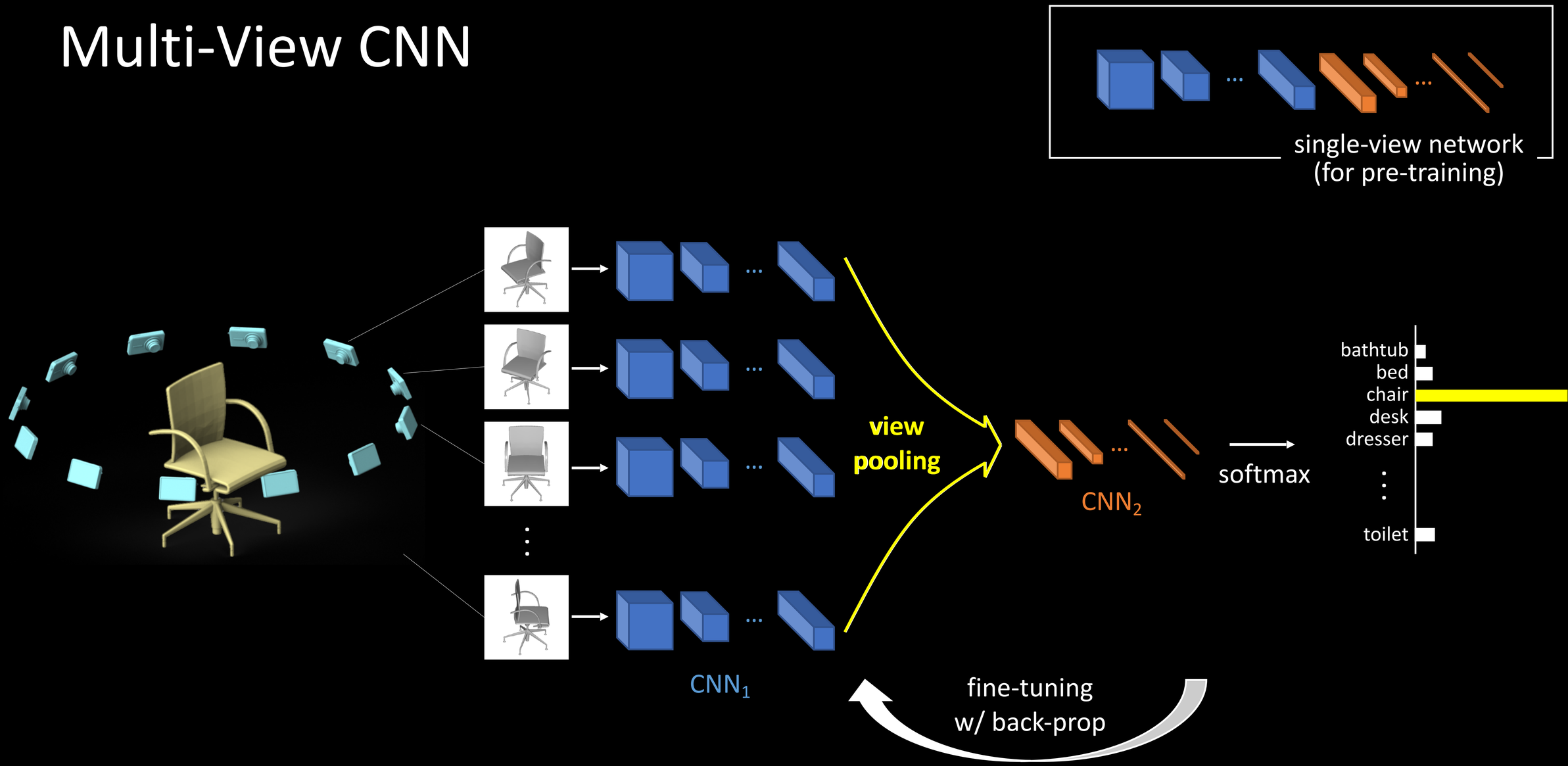
- Leverage powerful 2D architectures and pre-trained models
- Avoid expensive 3D operations
- Relate 3D shapes to images → enables flexible applications

Challenge:

- A list/set of descriptors is not convenient to use for learning
- In need of *a single compact descriptor* for each shape



Multi-View CNN



[ICCV '15] **H. Su**, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition".

Multi-View CNN

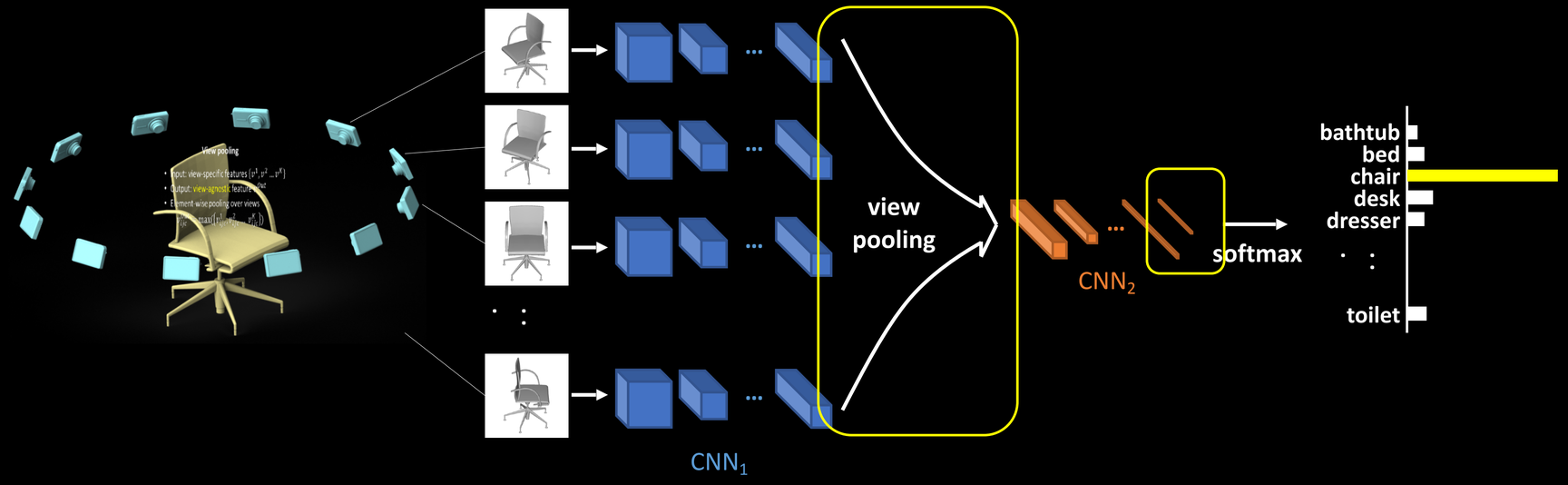
View pooling

- Input: view-specific features $\{v^1, v^2 \ldots v^K\}$
- Output: view-agnostic feature v^{Out}
- Element-wise pooling over views

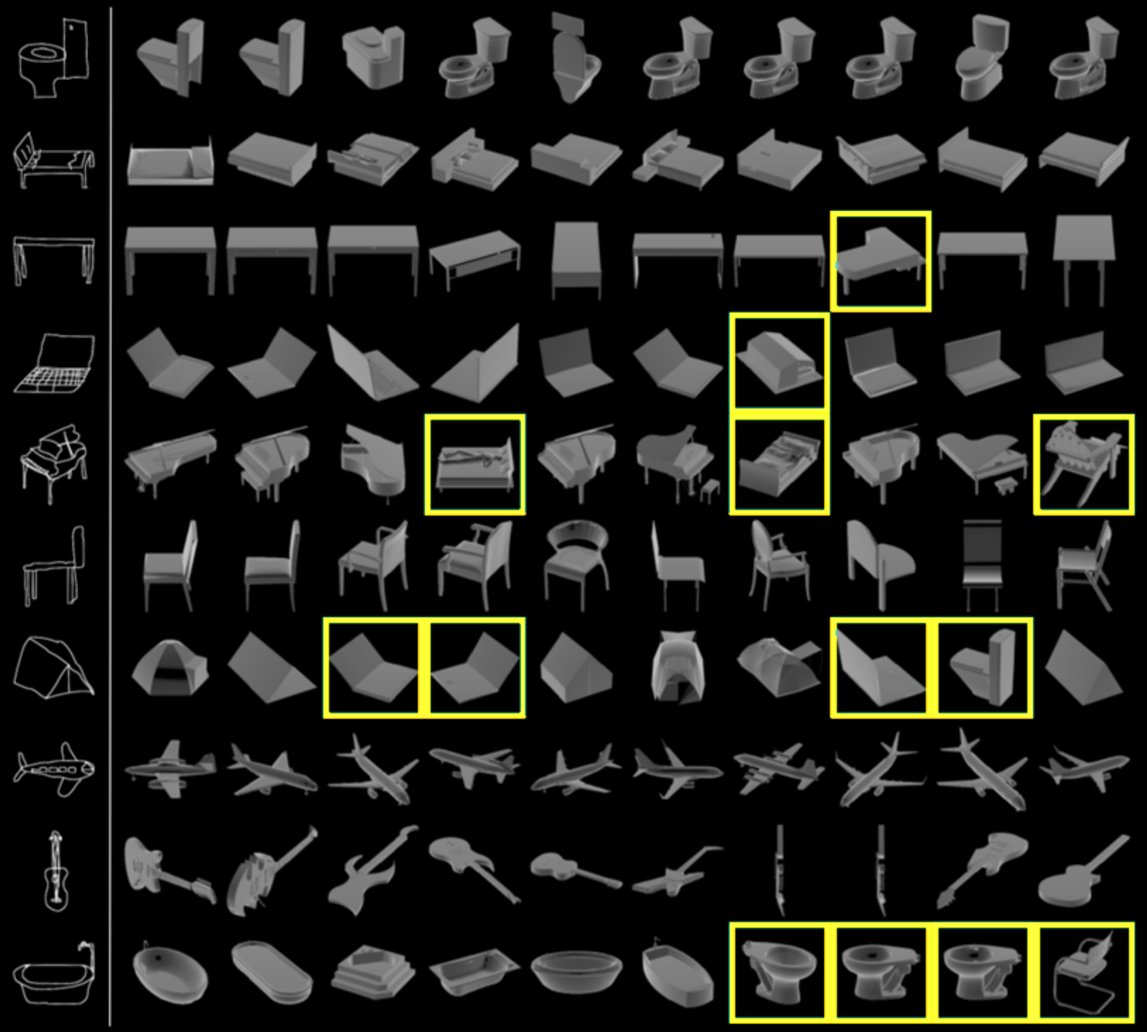
$$v_{ijc}^{Out} = \max(\{v_{ijc}^1, v_{ijc}^2 \ldots v_{ijc}^K\})$$

More compact shape descriptor

- Learn a low-rank metric $W \in \mathbb{R}^{128\times4096}$
- 4096-dim (fc_7 layer) → 128-dim
- Directly applicable to retrieval



Relating shapes to images



3D mesh

depth
buffer

Canny edge
detection

human
sketch

Multi-View CNN: Performance

State-of-the-art approach at the time of publishing:

- 75.5 (LFD) → 84.8 (FV) → 90.1 (MVCNN)
- Ranked #1 in SHREC '16 ShapeNet Retrieval Contest

Still a very competitive approach with some recent updates

- Multi-Resolution [1]
- Improved shading [2]
- Stronger backbone [2]

	Input	Per Class Acc. (%)	Per Inst. Acc. (%)
Our MVCNN	Images	**92.4**	**95.0**
Rotation Net		-	94.8
Dominant Set Clustering		-	93.8
MVCNN-MultiRes		91.4	93.8
MVCNN		90.1	90.1
DynamicGraph	Point Clouds	90.2	92.2
Kd-Networks		-	91.8
PointNet++		-	90.7
PointNet		86.2	89.2
VRN Single	Voxels	-	91.3
O-CNN		-	90.6
VoxNet		-	83.0
3DShapeNets		77.3	84.7
PointNet++	PC+Normal	-	91.9
FusionNet	Voxels+Images	-	90.8

Table from [2]

[1] C. Qi et al. Volumetric and Multi-View CNNs for Object Classification on 3D Data CVPR '16.
[2] J. Su et al. A Deeper Look at 3D Shape Classifiers. arXiv:1809.02560.

Multi-View CNN: Summary

- A collection of 2D views is highly informative for recognizing 3D objects.
- Using 2D CNNs allows leveraging powerful image architectures and available supervision.
- MVCNNs relate 3D shapes to 2D images, enabling cross-domain applications.

Volumetric CNN

Can we use CNNs without 2D-3D projection?

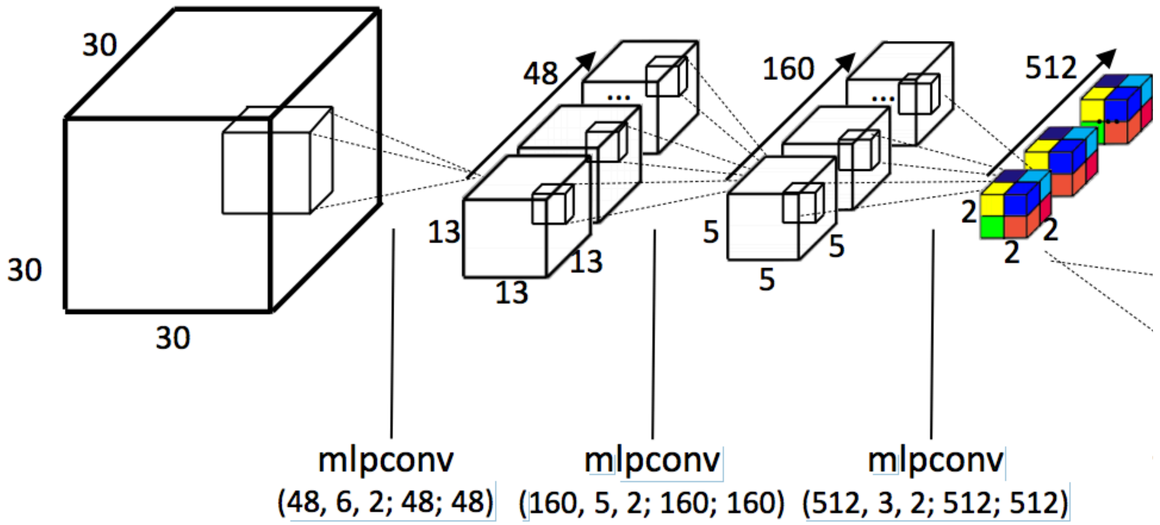
Yes, just use 3D convolutions!

Voxelization

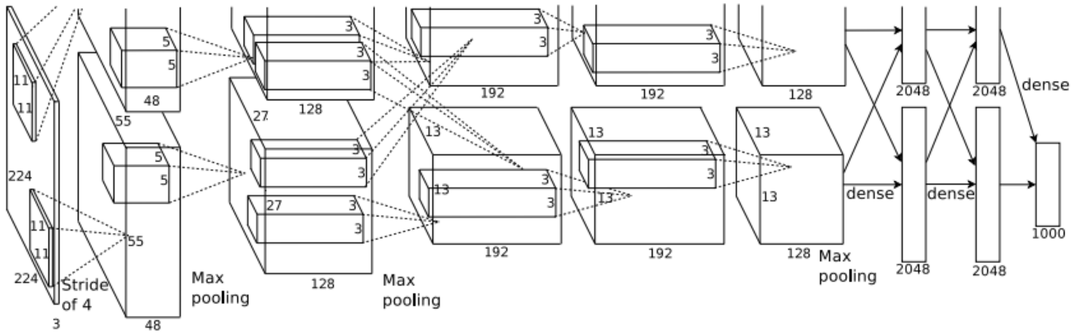
Represent the occupancy of regular 3D grids

3D CNN on Volumetric Data

3D convolution uses 4D kernels



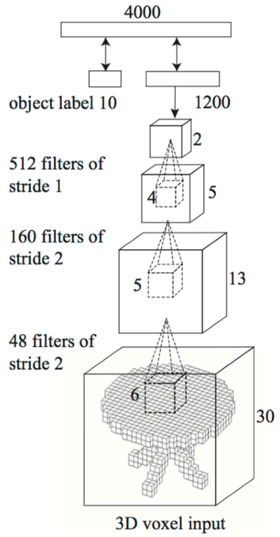
Complexity Issue



AlexNet, 2012

Input resolution: 224x224

224x224=50176

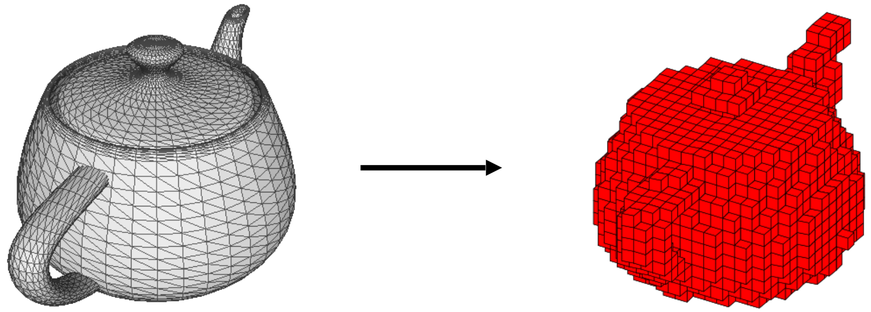


3DShapeNets, 2015

Input resolution: 30x30x30

224x224=27000

Complexity Issue

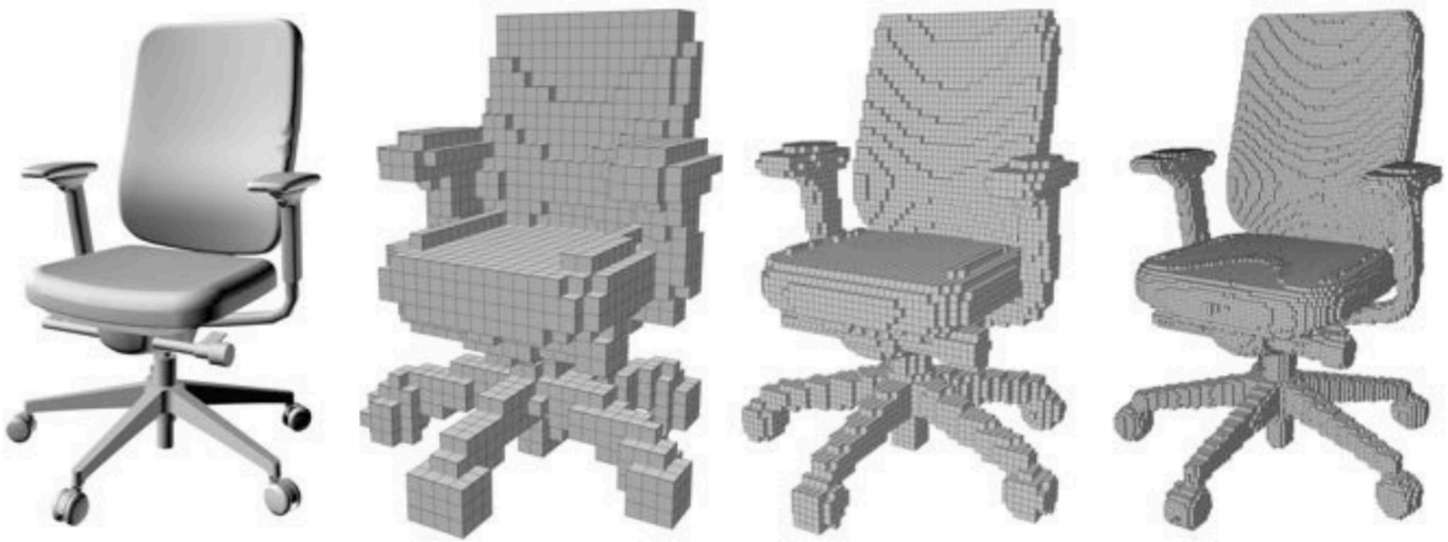


Polygon Mesh

Occupancy Grid
30x30x30

Information loss in voxelization

More Principled: Leverage Sparsity of 3D Surface Data

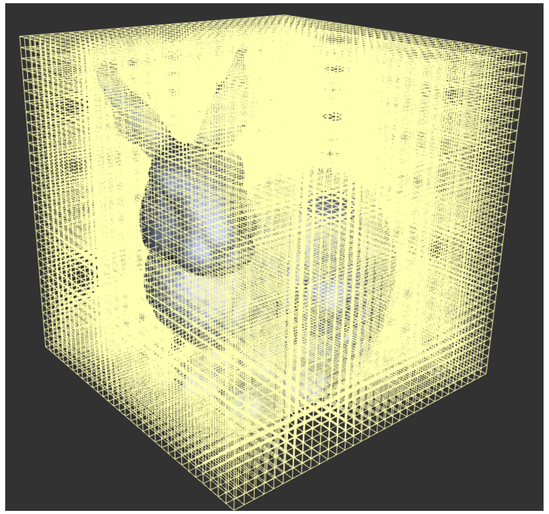


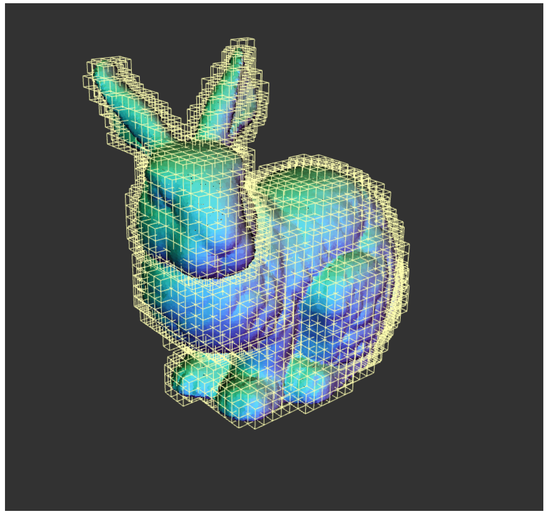
$\frac{\#occupied\ grid}{\#total\ grid}$

Occupancy:	10.41%	5.09%	2.41%
Resolution:	32	64	128

Store only the Occupied Grids

- Store the sparse surface signals
- Constrain the computation near the surface

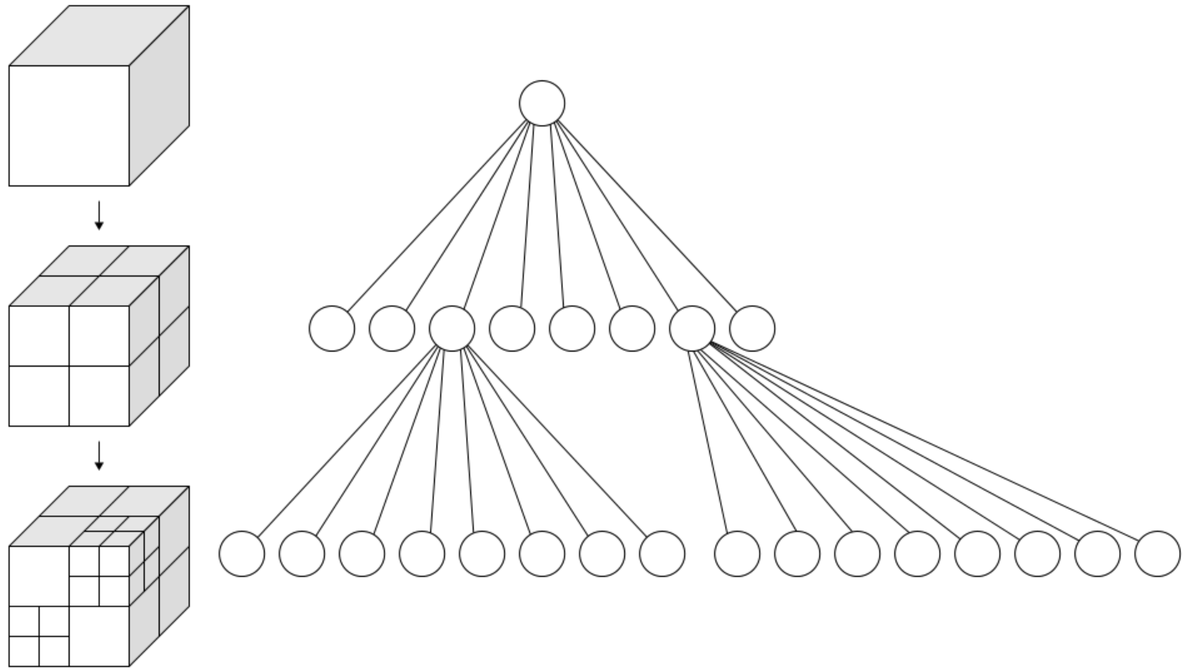




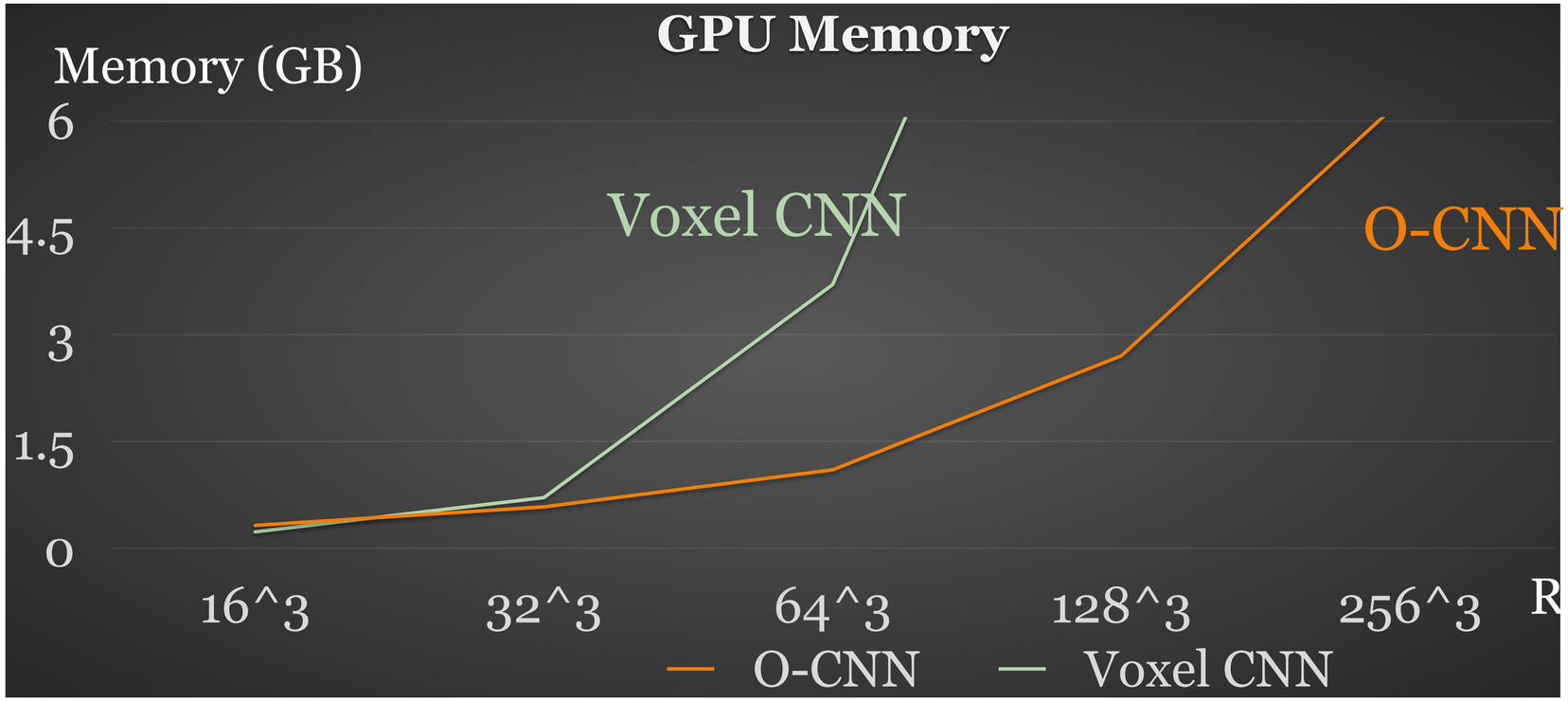
Octree: Recursively Partition the Space

Each internal node has exactly eight children

Neighborhood searching: Hash table

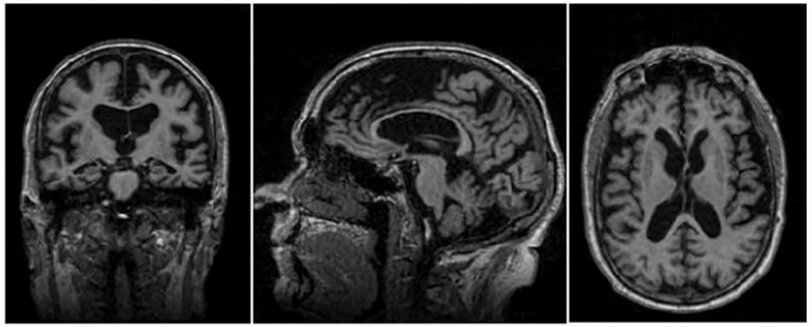


Memory Efficiency

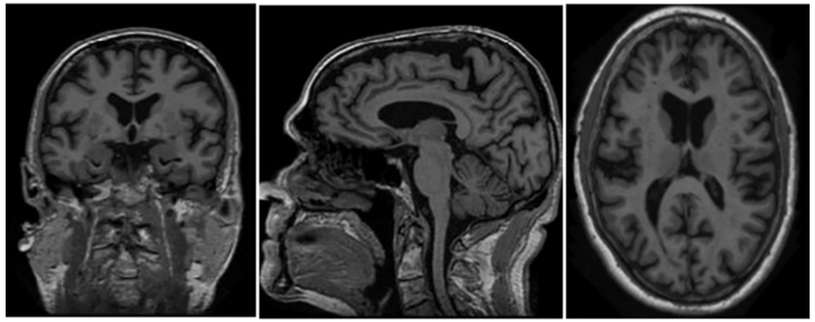


Sometimes the data is volumetric

CT Scan



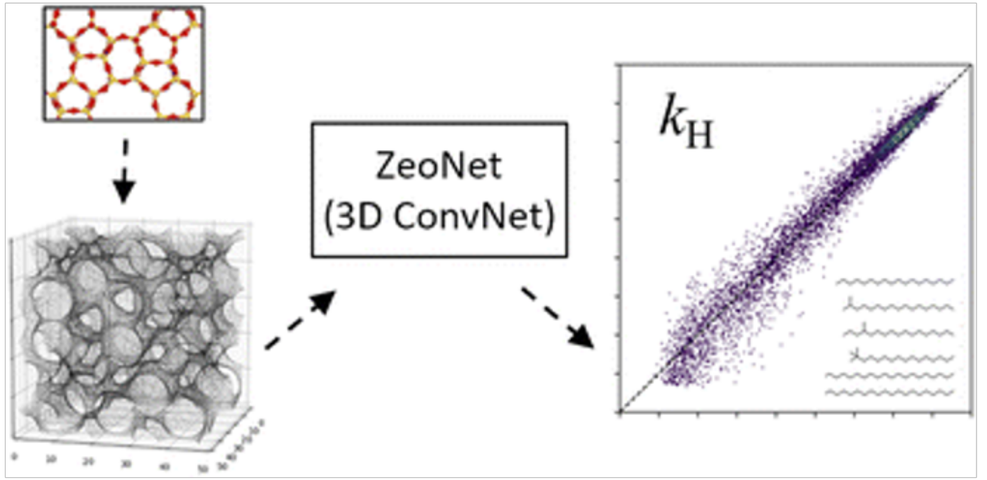
(a)



(b)

https://ic3.center.ufl.edu/news-events/medical-imaging/medical-imaging-datasets/

3D Materials (Zeolites)



Journal of Materials Chemistry A

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Check for updates

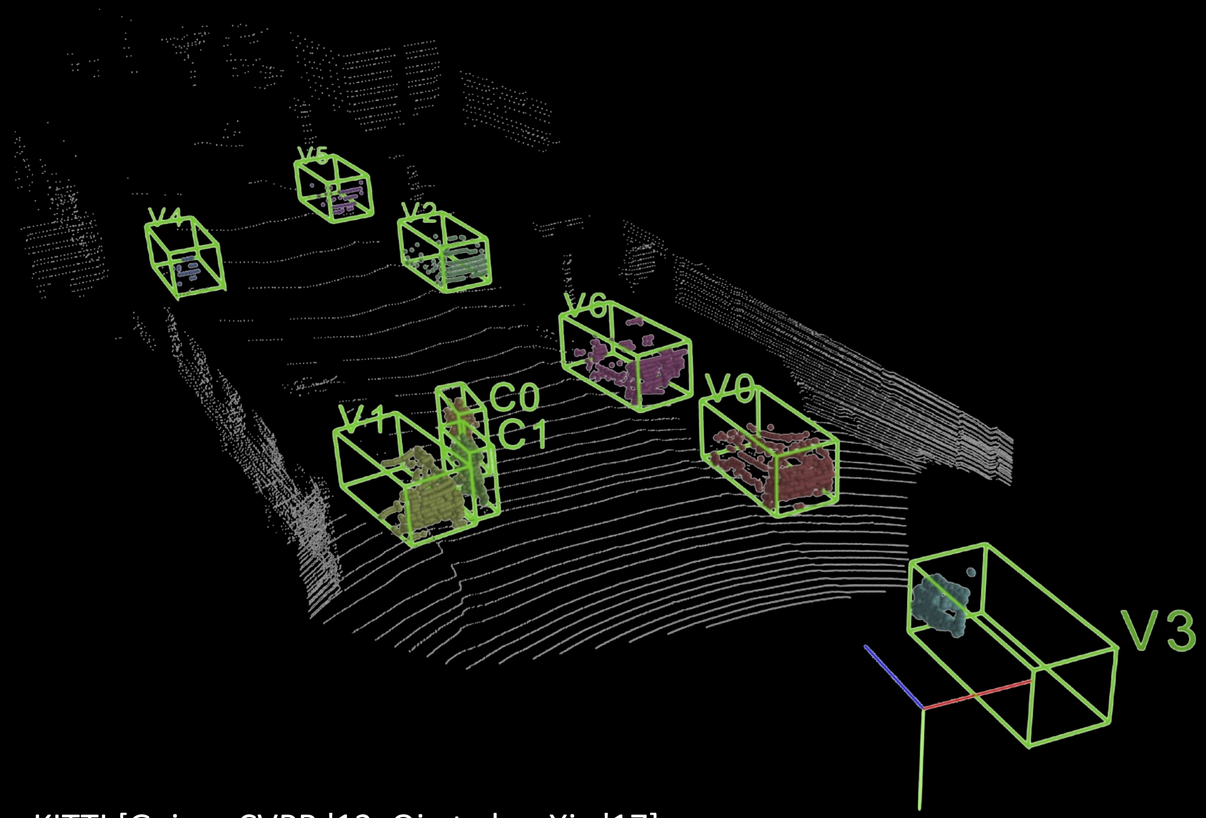
Cite this: J. Mater. Chem. A, 2023, 11, 17570

ZeoNet: 3D convolutional neural networks for predicting adsorption in nanoporous zeolites†

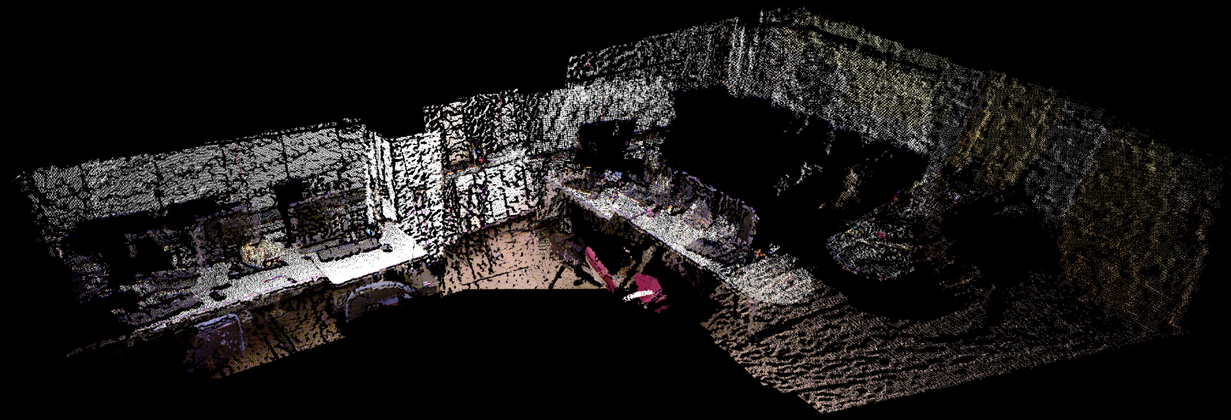
Yachan Liu,‡[a] Gustavo Perez,‡[b] Zezhou Cheng,[b] Aaron Sun,[b] Samuel C. Hoover,[a] Wei Fan,[a] Subhransu Maji*[b] and Peng Bai*[a]

3D volumetric models are very well suited!

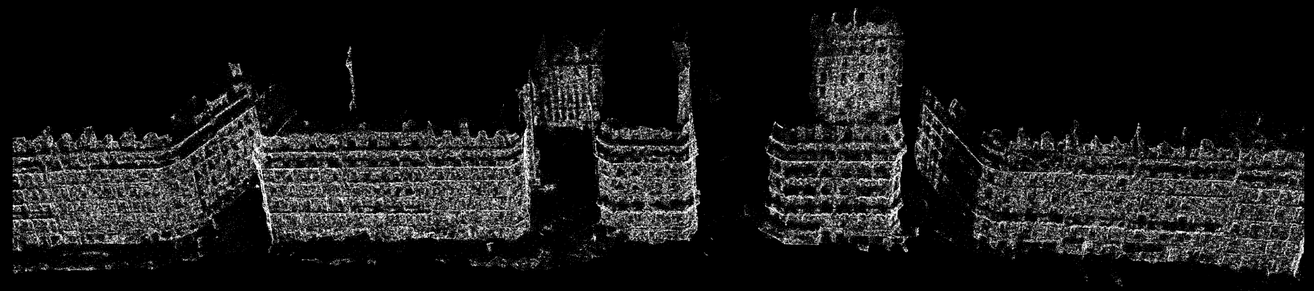
Point Networks



KITTI [Geiger CVPR '12, Qi et al. arXiv '17]



Cornell-RGBD [Koppula et al. NIPS '11]



Ruemonge2014 [Riemenschneider et al. ECCV '14]

Point clouds are highly sparse, and lack of grid structure.

Directly Process Point Cloud Data

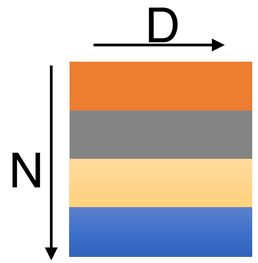
End-to-end learning for **unstructured,**

unordered point data

Qi, Charles R., et al. "**Pointnet: Deep learning on point sets for 3d classification and segmentation**", CVPR 2017
Zaheer, Manzil, et al. "**Deep sets**", NeurIPS 2017

Permutation invariance

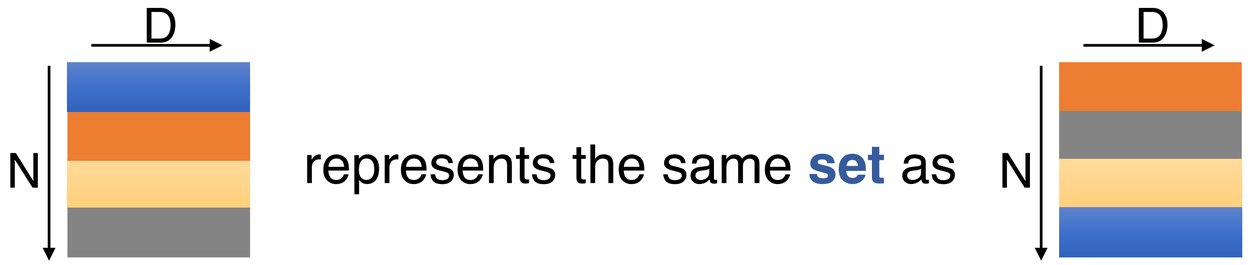
Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Permutation invariance

Point cloud: N **orderless** points, each represented by a D dim coordinate



2D array representation

Construct a Symmetric Function

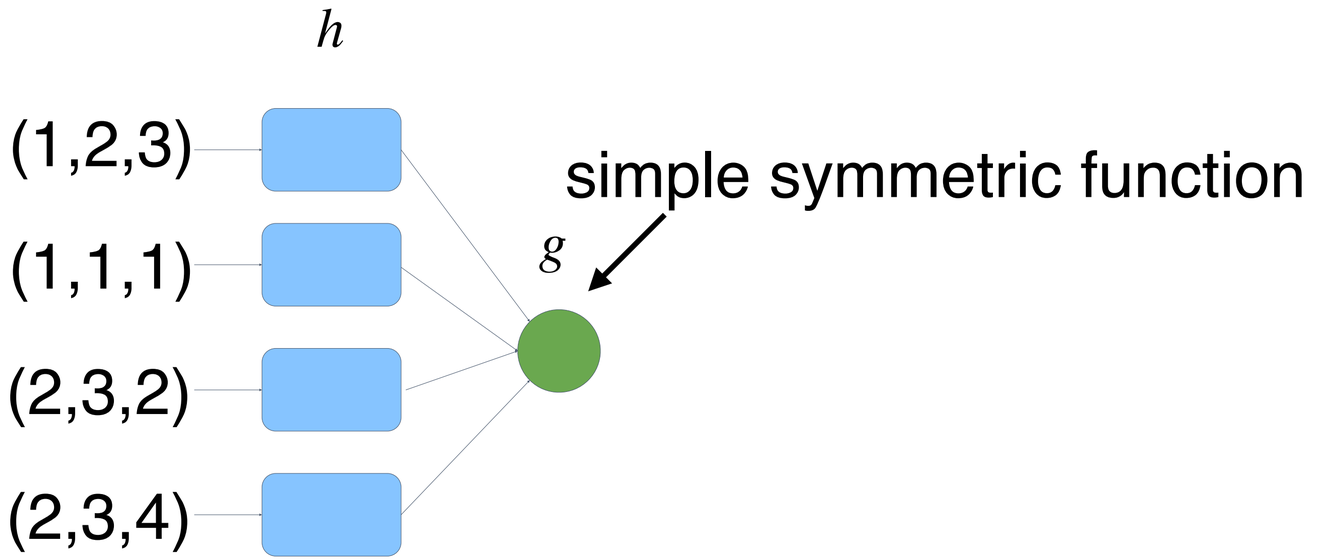
Observe:

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

Construct a Symmetric Function

Observe:

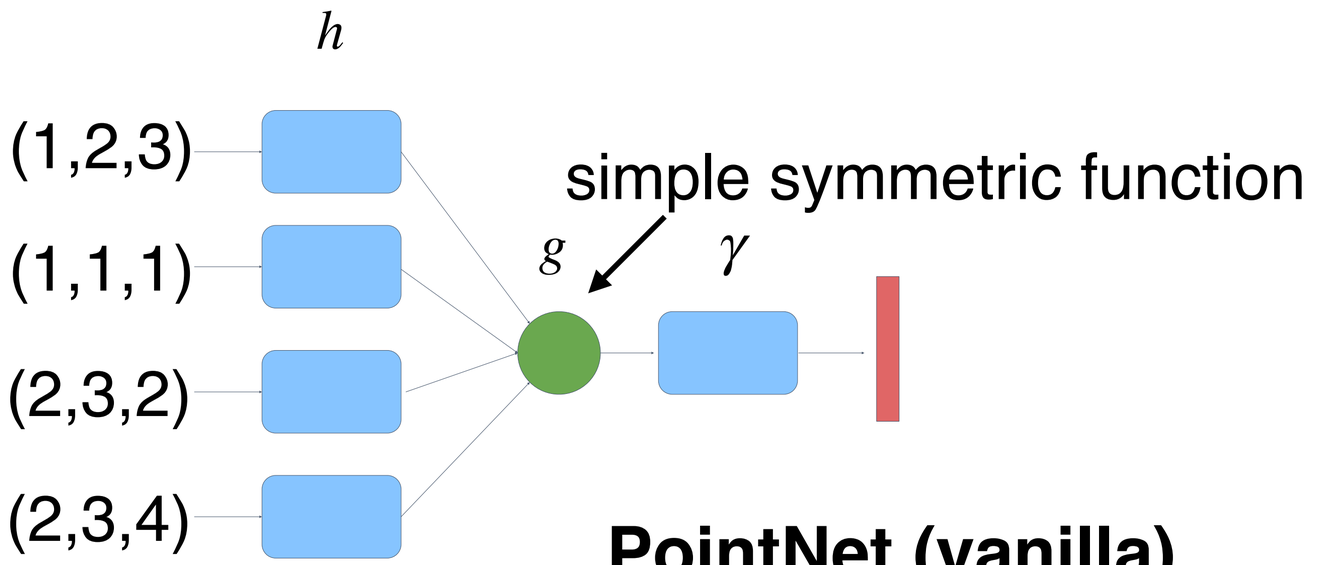
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

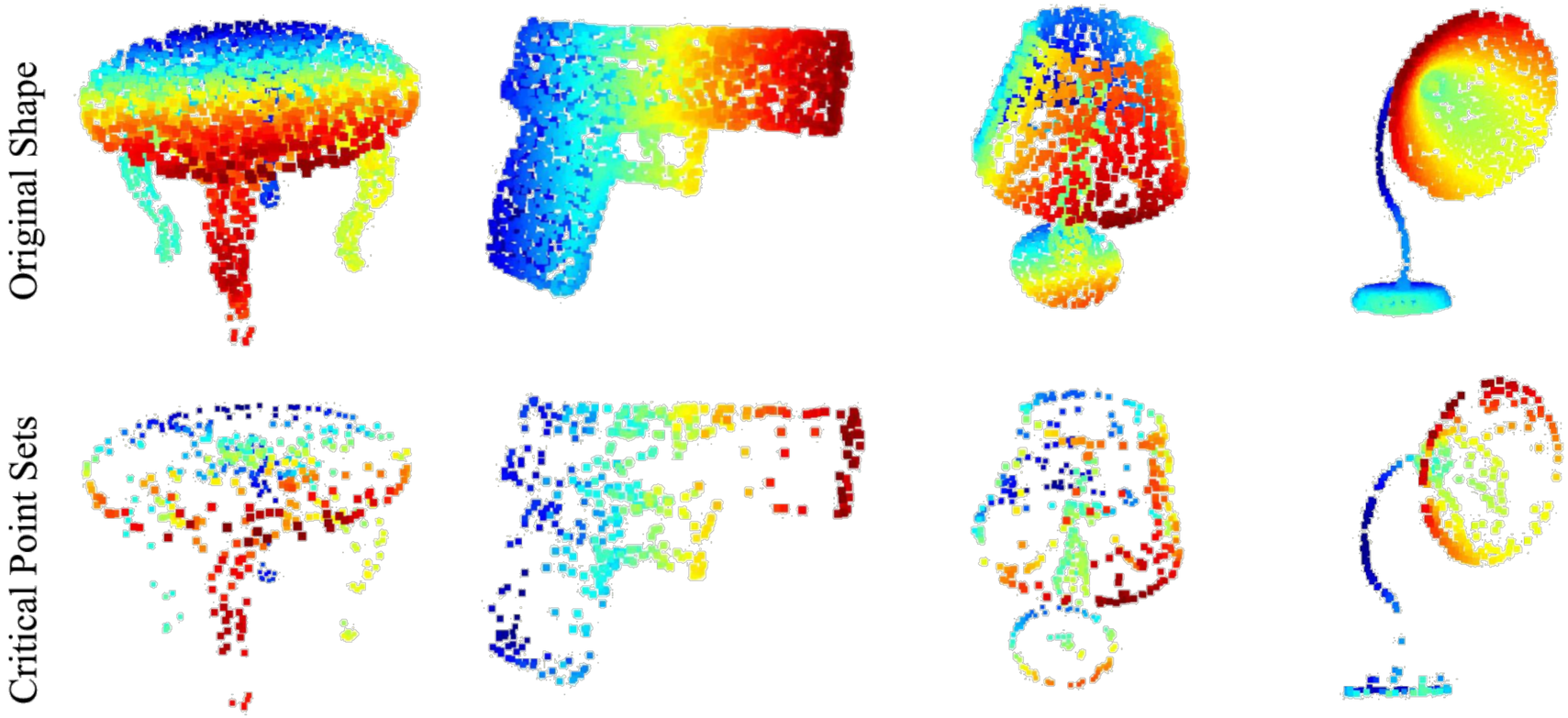
Observe:

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



PointNet (vanilla)

Visualize What is Learned by Reconstruction



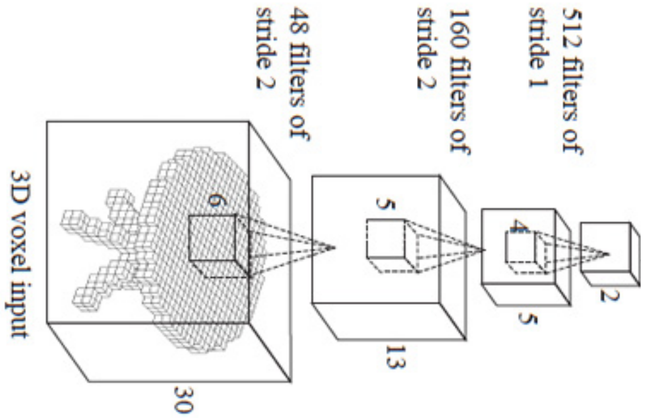
Salient points are discovered!

Limitations of PointNet

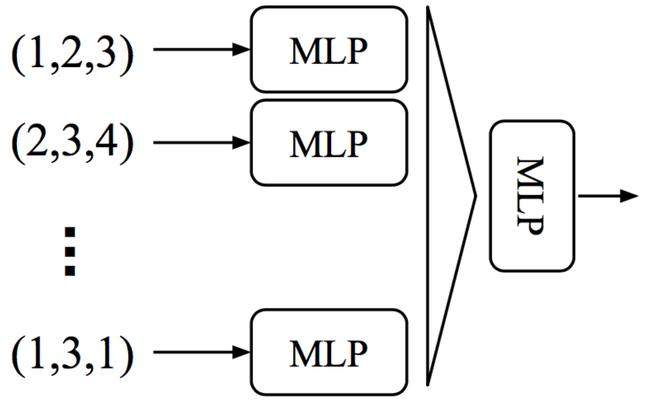
Hierarchical feature learning
Multiple levels of abstraction

v.s.

Global feature learning
Either one point or all points



3D CNN (Wu et al.)



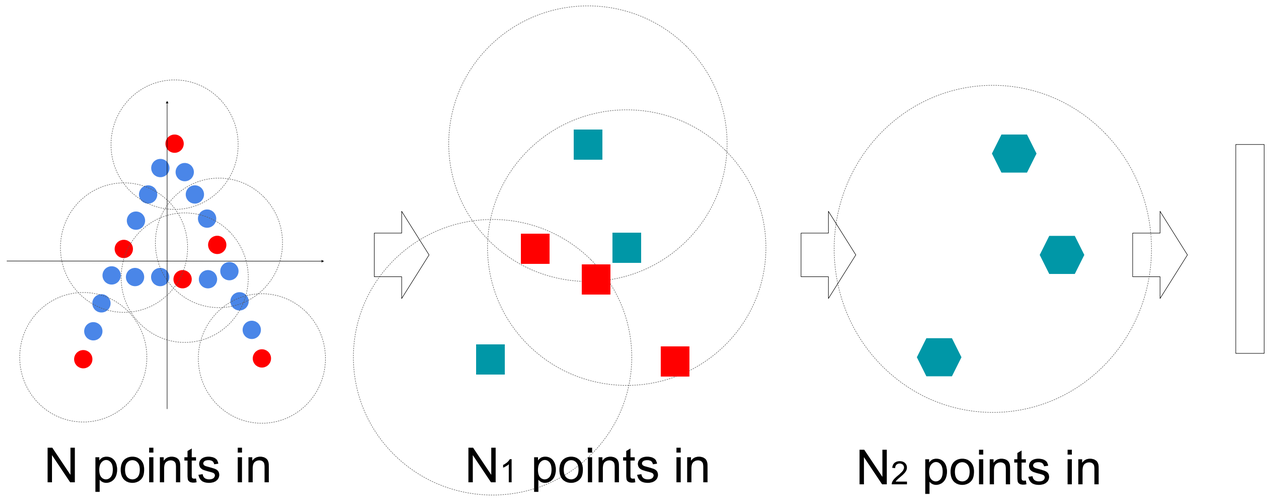
PointNet (vanilla) (Qi et al.)

- No local context for each point!
- Global feature depends on absolute coordinate. Hard to generalize to scene configurations!

Points in Metric Space

- Learn “kernels” in 3D space and conduct convolution
- Kernels have compact spatial support
- For convolution, we need to find neighboring points
- Possible strategies for range query
  - Ball query (results in more stable features)
  - k-NN query (faster)

PointNet v2.0: Multi-Scale PointNet



N points in (x,y)

N_1 points in (x,y,**f**)

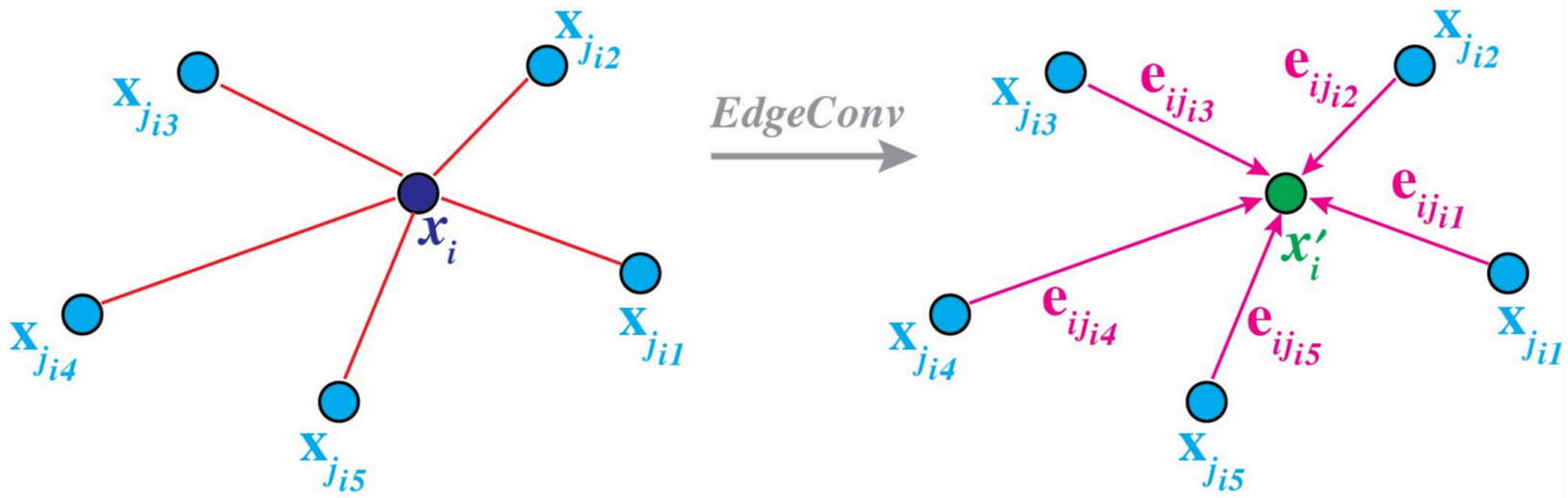
N_2 points in (x,y,**f'**)

Repeat

- Sample anchor points
- Find neighborhood of anchor points
- Apply PointNet in each neighborhood to mimic convolution

Point Convolution As Graph Convolution

- Points -> Nodes
- Neighborhood -> Edges
- Graph CNN for point cloud processing

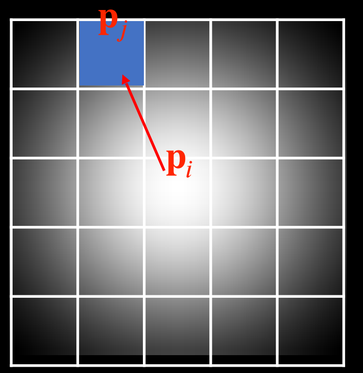


Wang et al., "**Dynamic Graph CNN for Learning on Point Clouds**", *Transactions on Graphics, 2019*

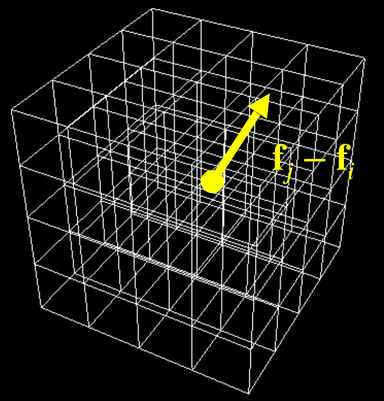
Liu et al., "**Relation-Shape Convolutional Neural Network for Point Cloud Analysis**", *CVPR 2019*

Spatial Convolution

$\mathbf{W}\left[\mathbf{p}_j - \mathbf{p}_i\right]$



$\mathbf{W}\left[\mathbf{f}_j - \mathbf{f}_i\right]$



Filter values based on position offsets w.r.t. center

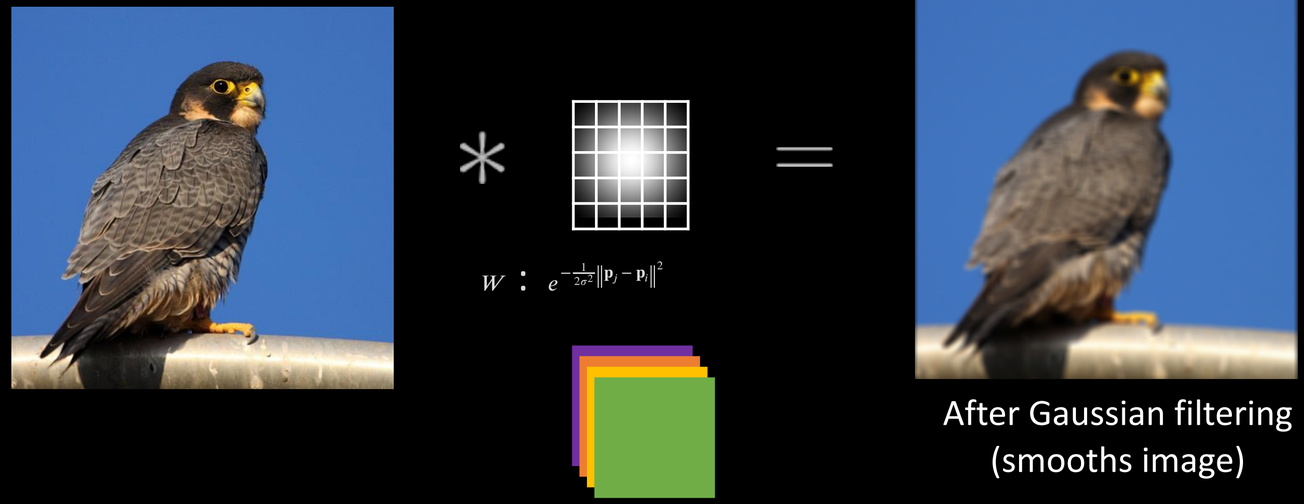
$$\mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}\left[\mathbf{p}_j - \mathbf{p}_i\right] \mathbf{v}_j + \mathbf{b}$$

output value

filter weights

position offset

input value



After Gaussian filtering (smooths image)

CNNs: w/ learned filters

High-Dimensional Filtering

$$\mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}[\mathbf{p}_j - \mathbf{p}_i]\mathbf{v}_j \longrightarrow \mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}[\mathbf{f}_j - \mathbf{f}_i]\mathbf{v}_j$$

positions

input value

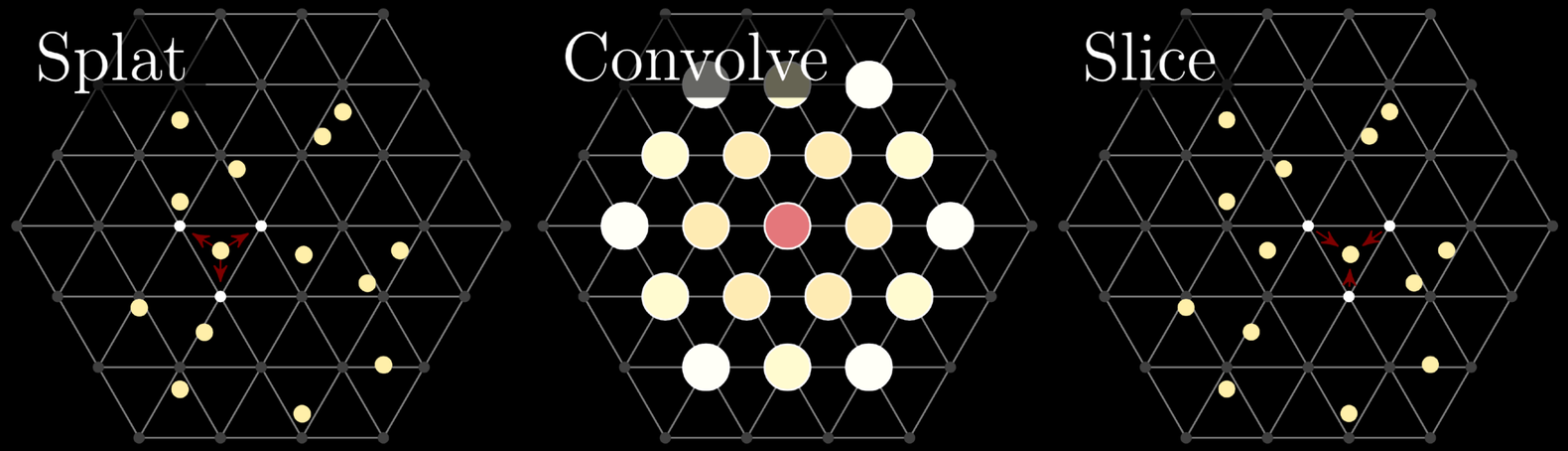
spatial convolution

high-dim. filtering

			Bilateral filter	Conv2d	
Image	pixel value $\mathbf{v}$	(r, g, b)	(r, g, b)	$\mathbf{v}(\dots)$	$\mathbf{v}(\dots)$
	pixel position $\mathbf{f}$	(x, y)	(x, y, r, g, b)	(x, y)	$(x, y, depth)$
3D Point cloud	point value $\mathbf{v}$	1	(x, y, z)	(r, g, b)	$\mathbf{v}(\dots)$
	point position $\mathbf{f}$	(x, y, z)	(x, y, z)	(x, y, z, n_x, n_y, n_z)	(x, y, z)

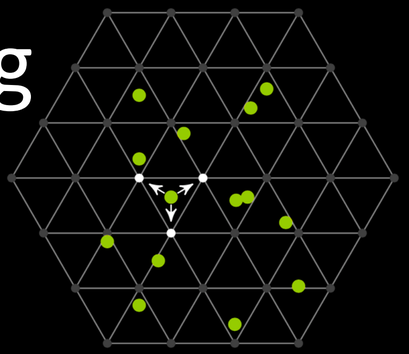
Efficient Sparse High-Dimensional Filtering

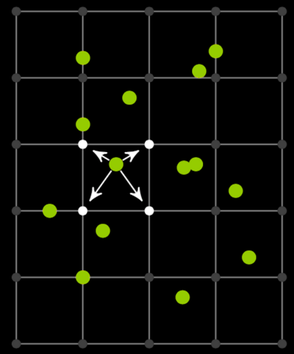
Bilateral Convolution Layer (BCL) [1,2,3]



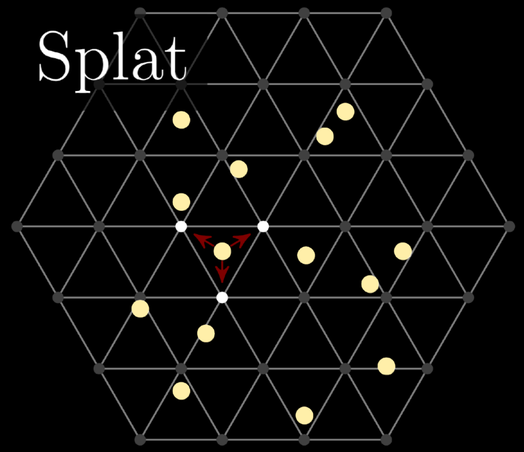
[1] A. Adams, J. Baek and M. A. Davis. Fast High-Dimensional Filtering Using the Permutohedral Lattice. Computer Graphics Forum '10
[2] V. Jampani, M. Kiefel and P. V. Gehler. Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks. CVPR '16
[3] M. Kiefel, V. Jampani and P. V. Gehler. Permutohedral Lattice CNNs. ICLR '15 workshops

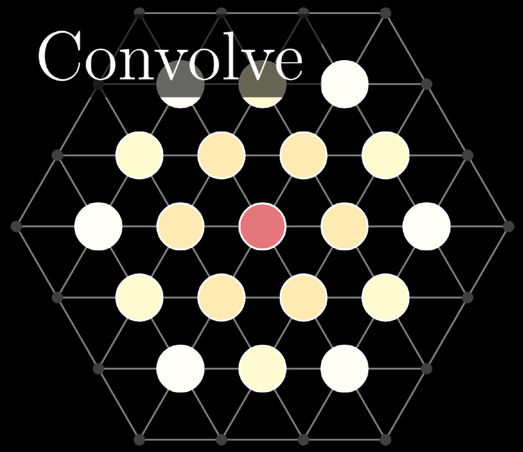
Efficient Sparse High-Dimensional Filtering

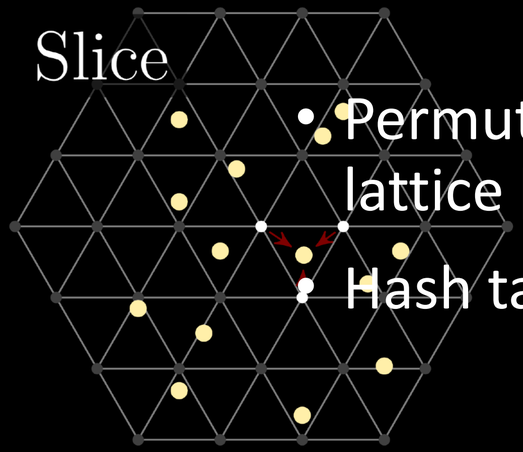




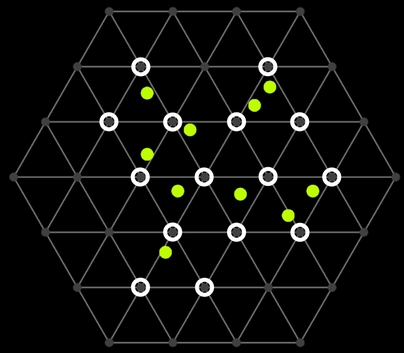
Data structures for efficient computation







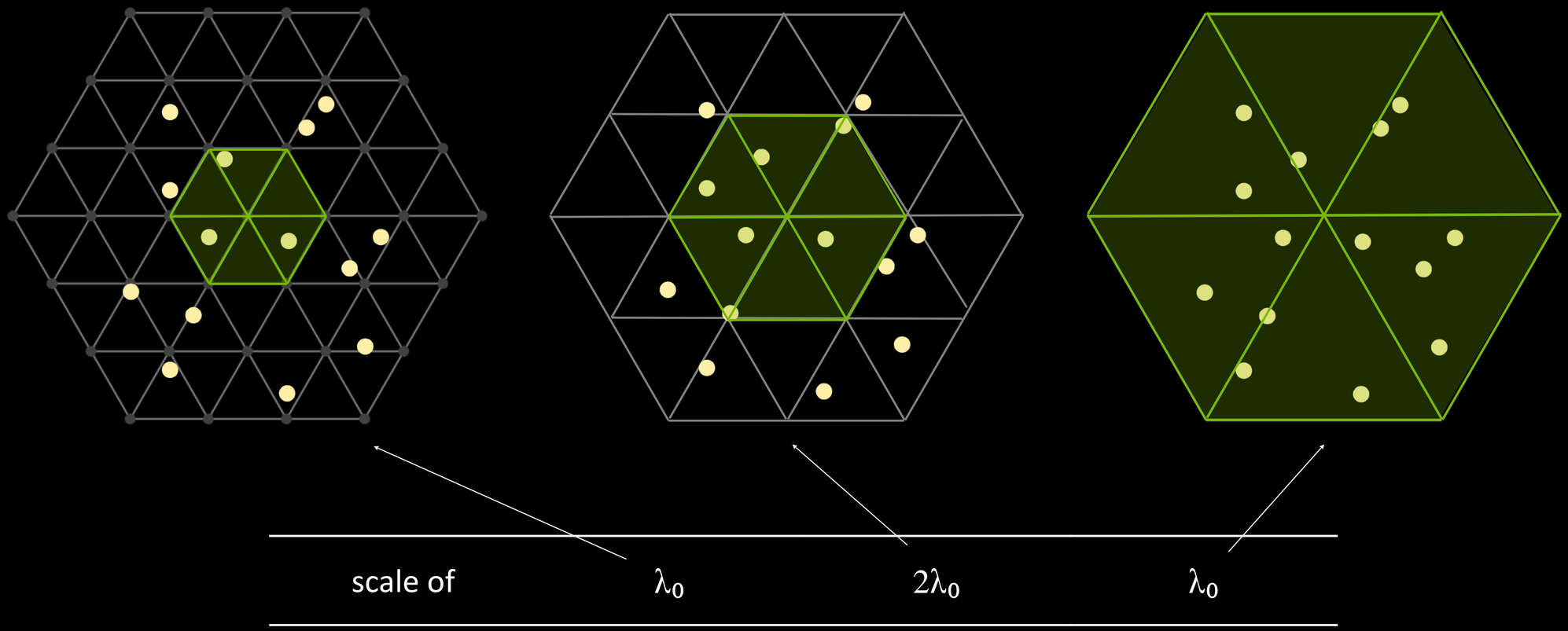
- Permutohedral lattice [1]
- Hash table

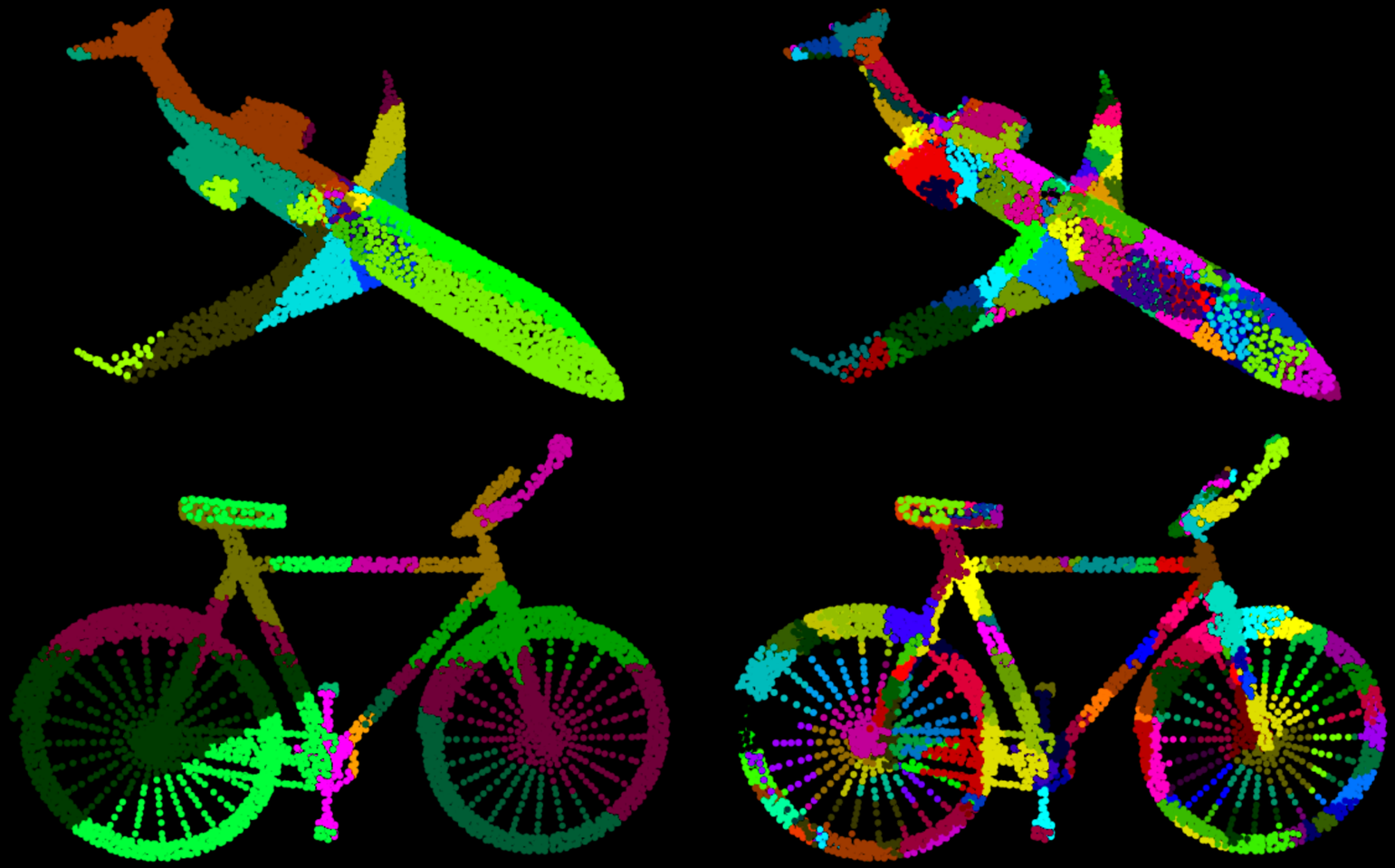


[1] Adams *et al.* Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum '10

Controlling receptive fields

$$\mathbf{v}_i' = \sum_{j \in \Omega(i)} \mathbf{W}\left[\mathbf{f}_j - \mathbf{f}_i\right]\mathbf{v}_j$$





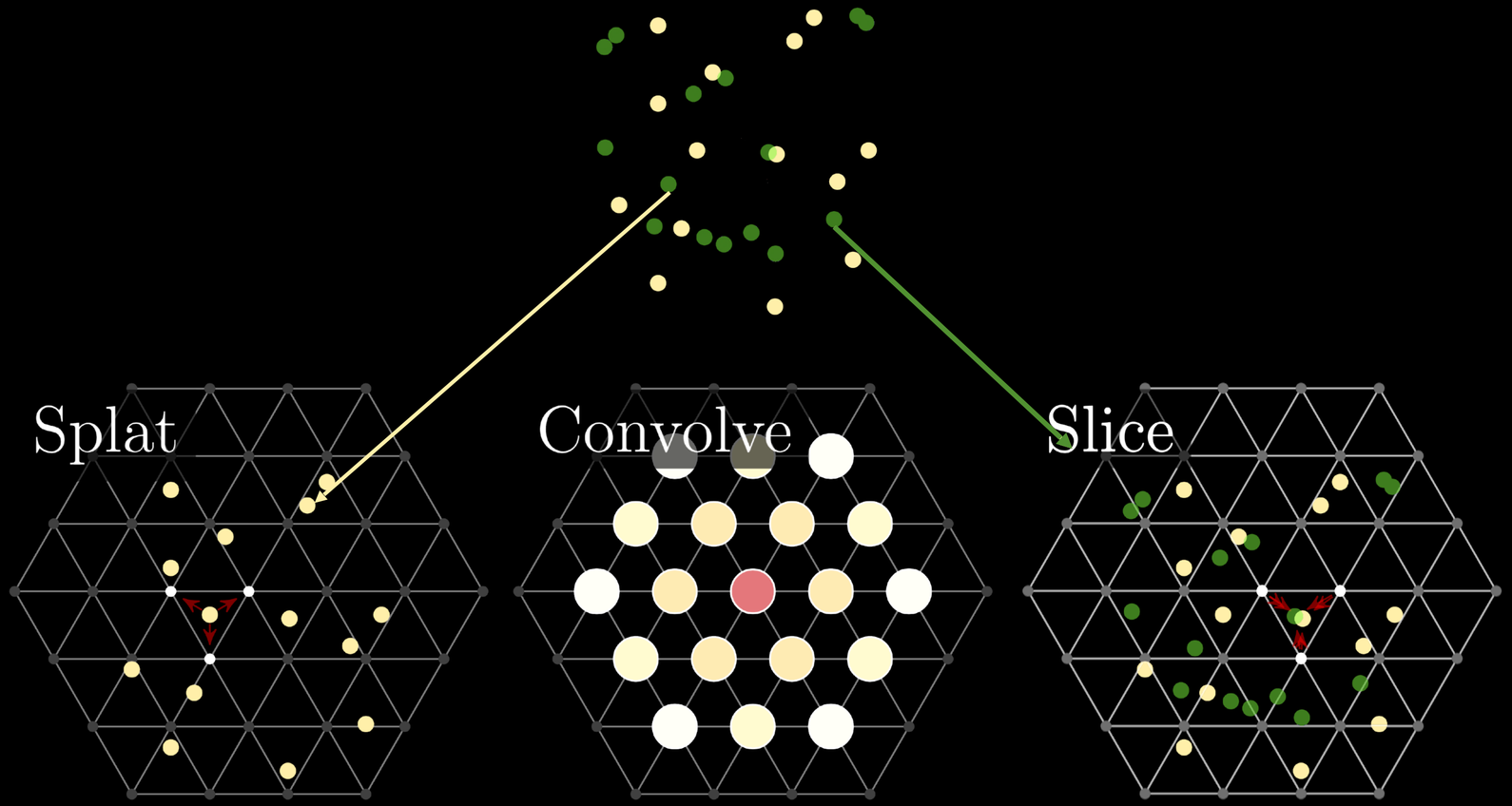
lattice feature: (x, y, z)

lattice scale: $8 \cdot \lambda_0$

lattice feature: (x, y, z)

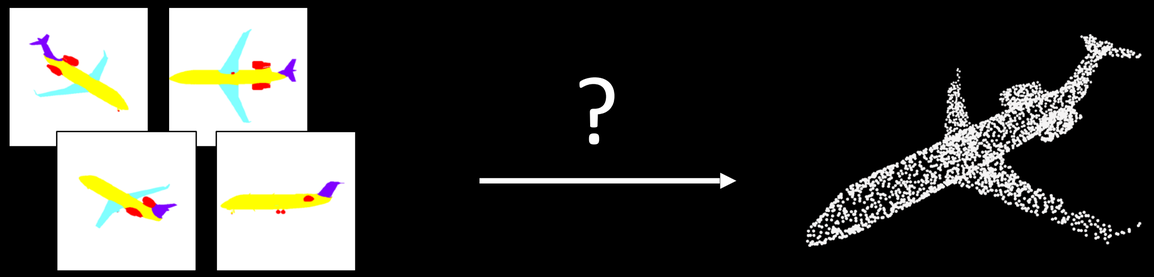
lattice scale: λ_0

Input and output can be at different points



Smooth projection between 2D and 3D

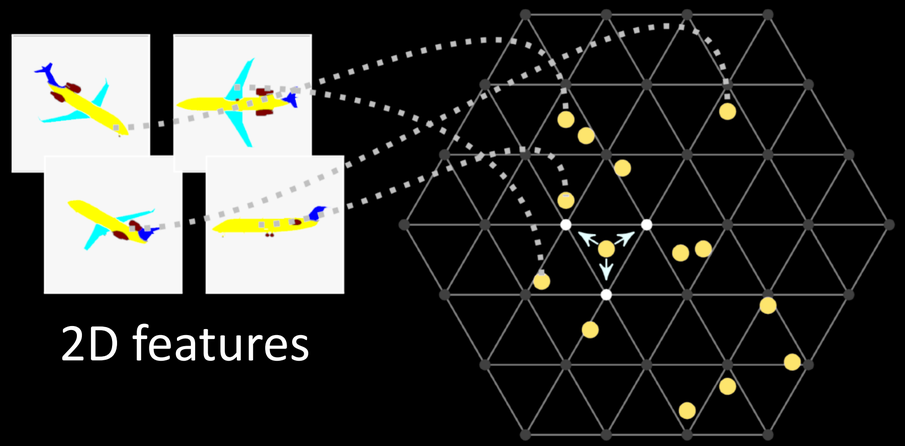
- Each 2D pixel p: $f(p)$, (x, y, z)



2D features

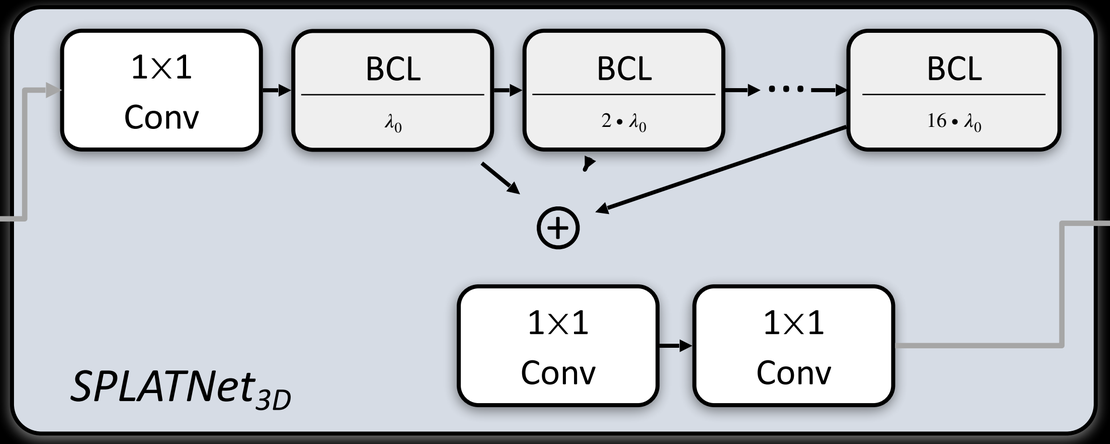
Smooth projection between 2D and 3D

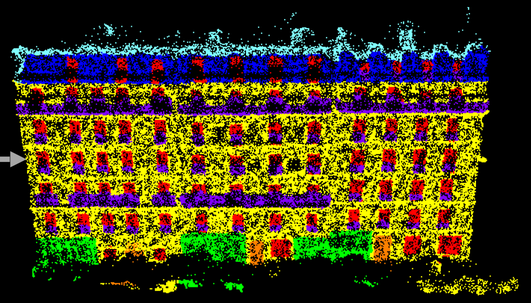
- Each 2D pixel p: $f(p)$, (x, y, z)
  → Point value: $f(p)$
  → Point position: (x, y, z)



SPLATNet Architectures

Input 3D point cloud

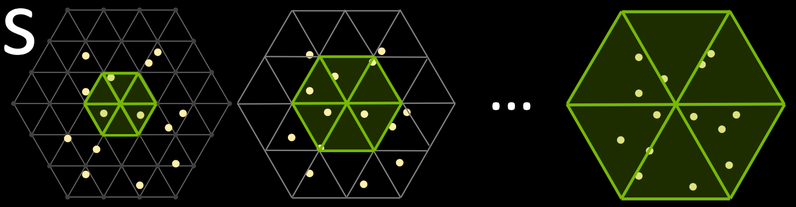




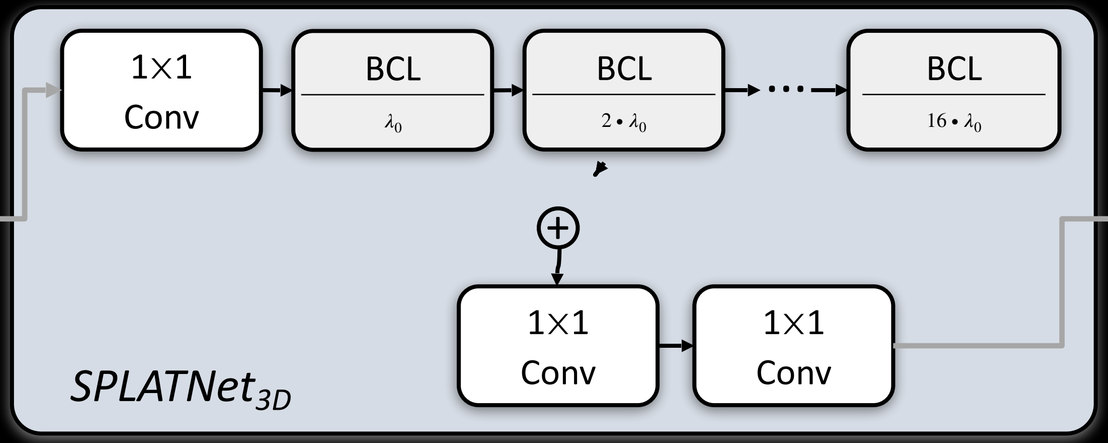
3D prediction

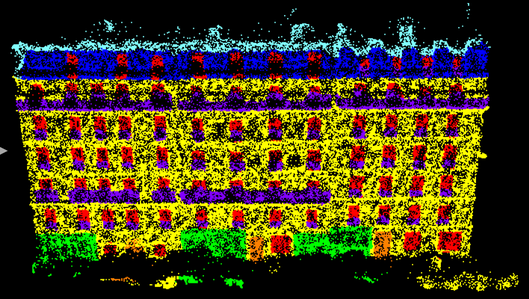
[CVPR '18] **H. Su**, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, MH Yang, and J. Kautz, “SPLATNet: Sparse Lattice Networks for Point Cloud Processing”. [BEST PAPER HONORABLE MENTION]

SPLATNet Architectures



Input 3D point cloud



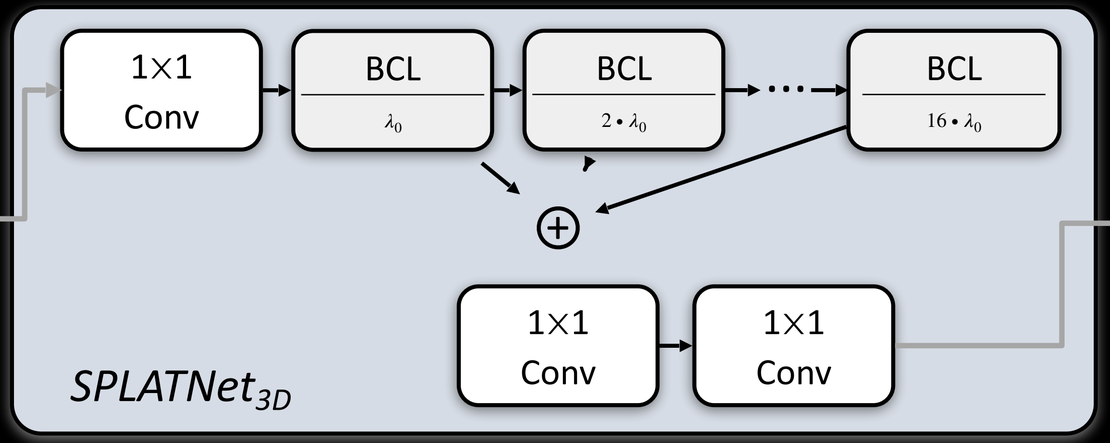


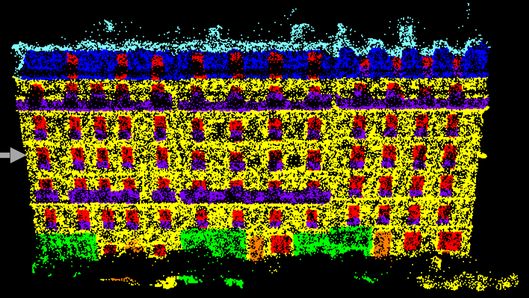
3D prediction

⊕ concatenation

SPLATNet Architectures

Input 3D point cloud



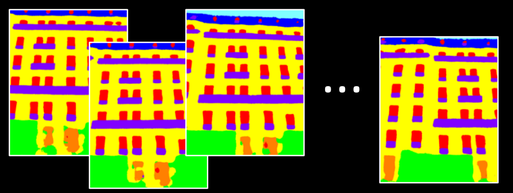


3D prediction

Input images

BCL
2D → 3D

BCL
3D → 2D

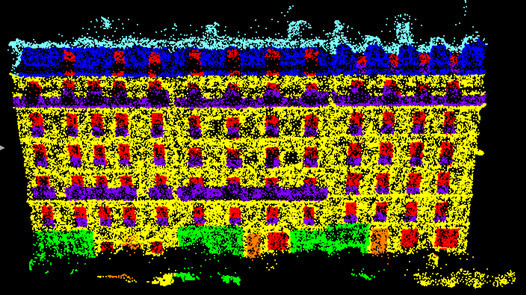


2D predictions

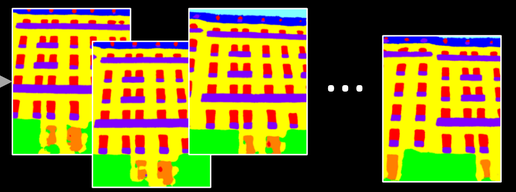
SPLATNet Architectures

Input 3D point cloud

Input images



3D prediction



2D predictions

Facade Segmentation (Ruemonge2014 [1])

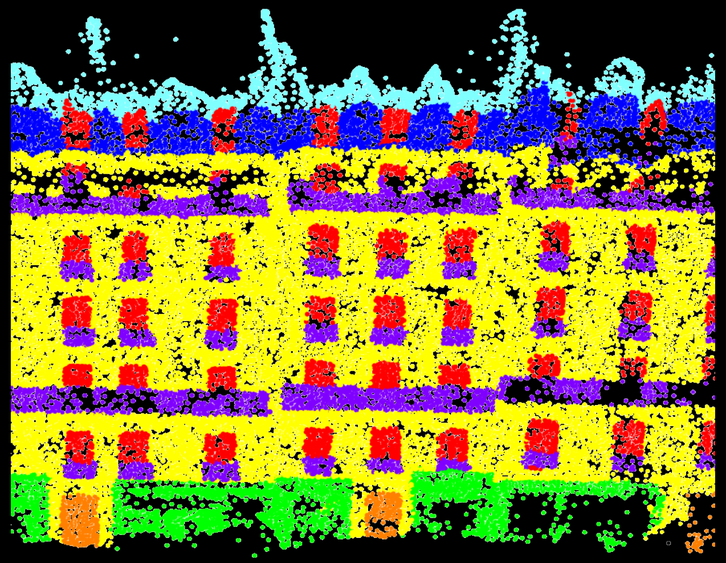
with only 3D data

	IoU	runtime (min)
OctNet [2]	59.2	-
Autocontext$_{3D}$ [3]	54.4	16
SPLATNet$_{3D}$	**65.4**	**0.06**

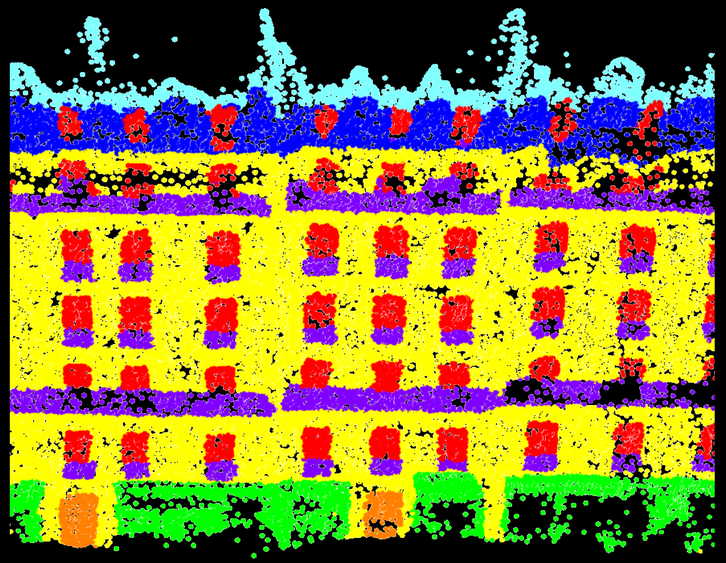
with both 2D & 3D data

	IoU	runtime (min)
Autocontext$_{2D-3D}$ [3]	62.9	87
SPLATNet$_{2D-3D}$	**69.8**	**1.2**

Input point cloud



Ground-truth



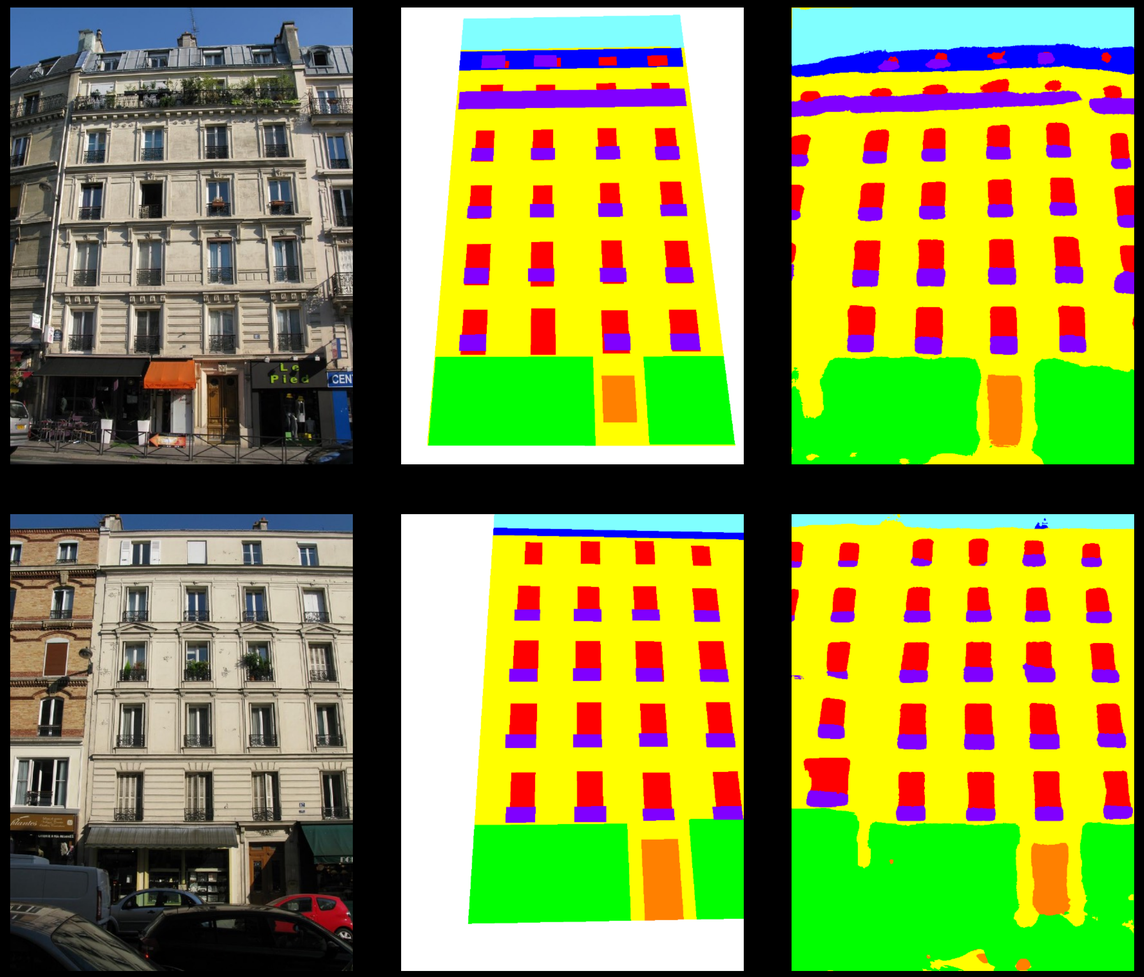
Ours (*SPLATNet$_{2D-3D}$*)

[1] Riemenschneider et al. ECCV '14 [2] Riegler *et al.* CVPR '17 [3] Gadde *et al.* PAMI '17

2D Predictions

	IoU	runtime (min)
Autocontext$_{2D}$ [1]	60.5	11.7
Autocontext$_{2D-3D}$ [1]	62.7	14.6
2D CNN [2] only	**69.3**	**0.4**
SPLATNet$_{2D-3D}$	**70.6**	**4.3**

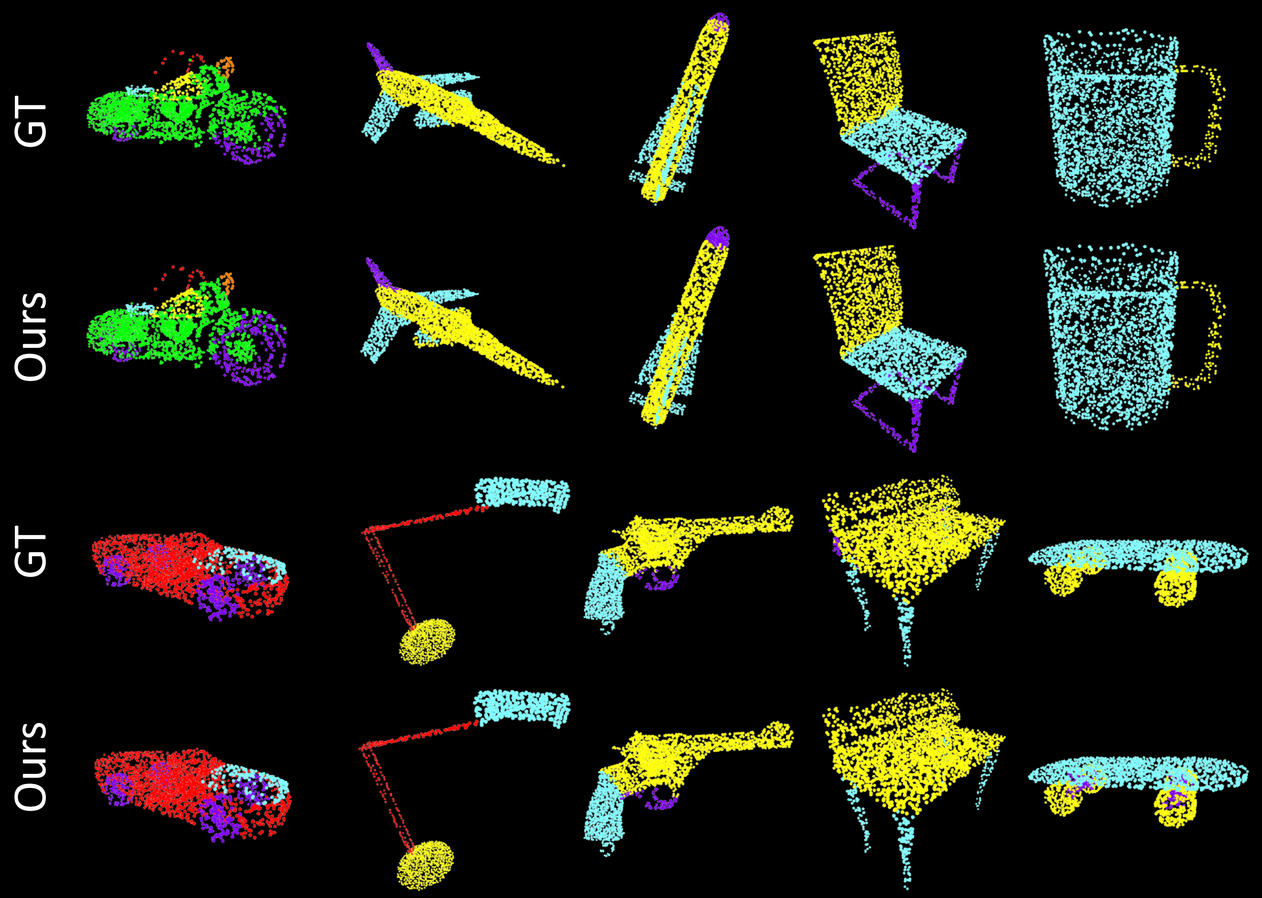
[1] Gadde *et al.* PAMI '17
[2] Chen *et al.* ICLR '15



Input image | Ground-truth | Our prediction

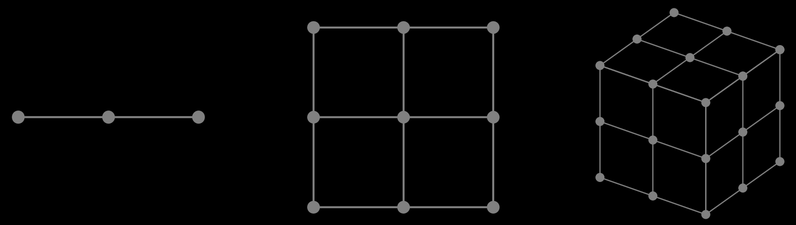
3D Object Part Labeling (ShapeNet [1])

	class avg. IoU	instance avg. IoU
Yi *et al.* [1]	79.0	81.4
3DCNN [2]	74.9	79.4
Kd-network [3]	77.4	82.3
PointNet [2]	80.4	83.7
PointNet++ [4]	81.9	85.1
SyncSpecCNN [5]	82.0	84.7
SPLATNet$_{3D}$	**82.0**	**84.6**
SPLATNet$_{2D-3D}$	**83.7**	**85.4**



[1] Yi *et al.* SIGGGRAPH Asia '16
[2] Qi *et al.* CVPR '17
[3] Klokov and Lempitsky ICCV '17
[4] Qi *et al.* NIPS '17
[5] Yi *et al.* CVPR '17

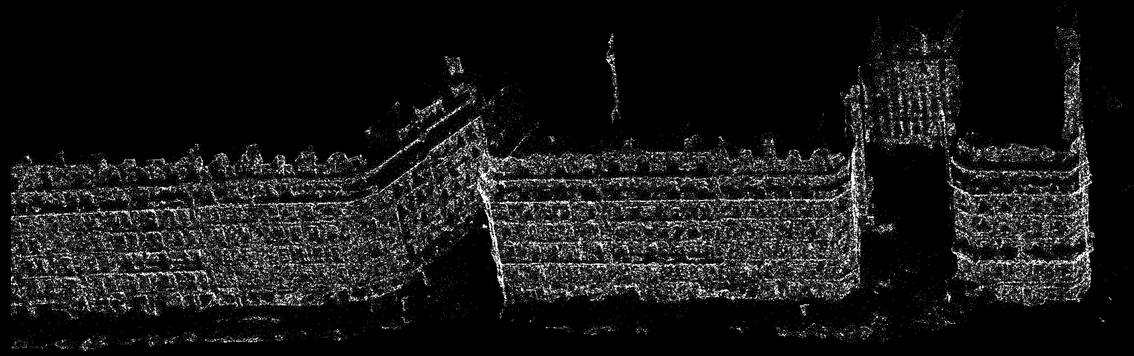
Remarks



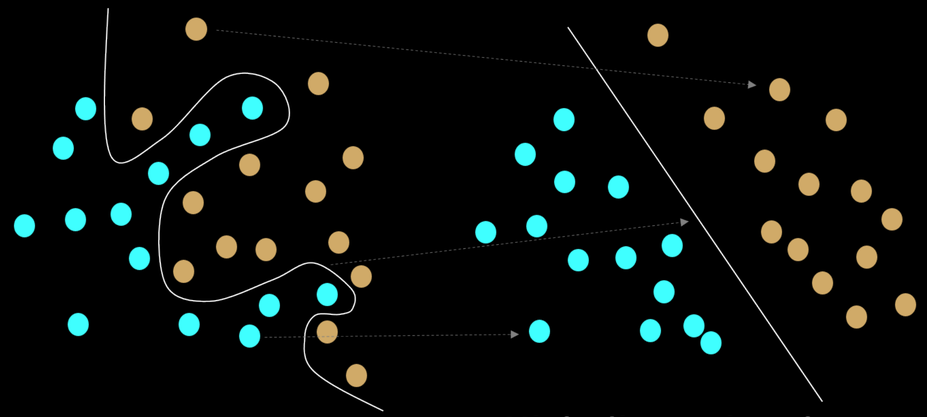
1-D 2-D 3-D 4-D, 5-D, 6-D, etc.

Equip neural networks with a richer class of sparse high-dimensional operations

1. Data that are inherently sparse and high-dimensional
   E.g. 3D data, BRDF material properties, video
2. Project data to higher dimensions
   E.g. bilateral filter

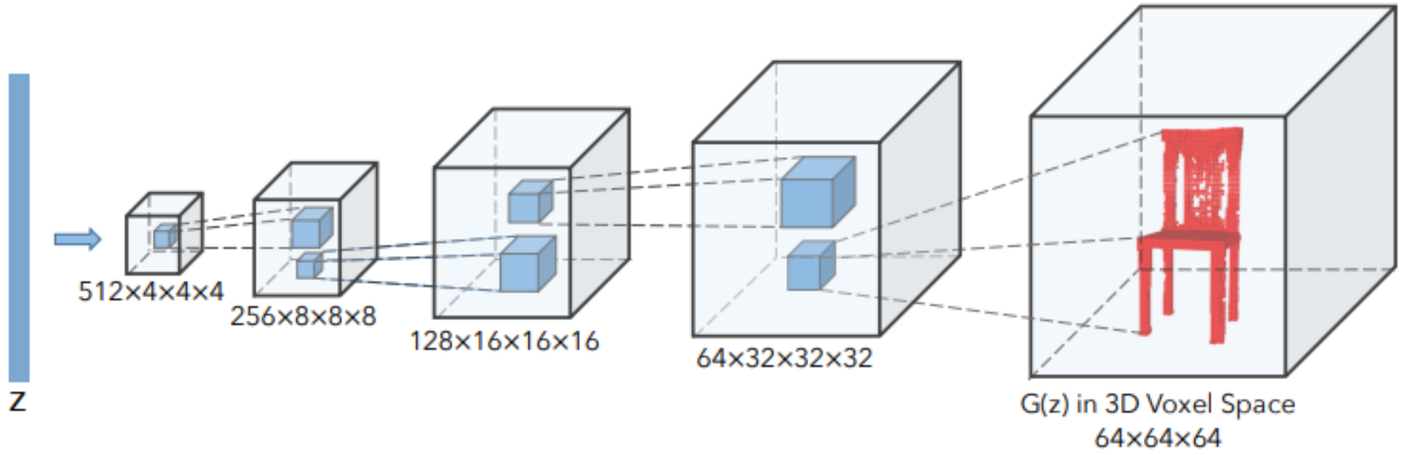


Ruemonge2014 [Riemenschneider et al. ECCV '14]



Input Space High dimensional space

Volumetric Generation

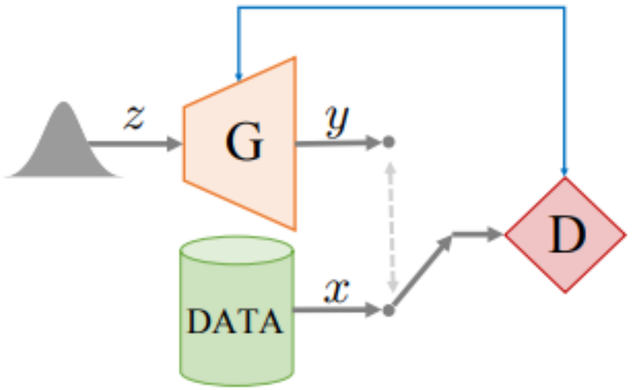


$$logD(x) + log(1 - D(G(z)))$$

Wu et al., “**Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling**”, *NeurIPS 2016*

Point Cloud Generation

- FC layer as generator
- PointNet as discriminator
- WGAN



Achlioptas et al., “**Learning Representations and Generative Models for 3D Point Clouds**”, *ICML 2018*

Many Issues

- Still cannot generate high quality local details
- Still hard to generate complex structures
- Partly limited by 3D training data, which is much smaller than images

Recent trends

Image (s) to 3D

- **Task:** Given input image(s), estimate the underlying 3D structure.
- **Scale ambiguity:** Outputs are often estimated only up to an unknown scale

- **Monocular depth estimation:** Single image to depth
  - Early models were ConvNet-based and trained on small datasets
  - Recent models are based on diffusion models and transformers, and are trained on massive datasets.

- **Multi-view 3D:** Multiple images as input; simultaneously estimate camera poses and the underlying geometry for a more complete 3D reconstruction
- Recent methods are often transformer-based and trained on massive datasets

Monocular depth estimation

Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation

Bingxin Ke Anton Obukhov Shengyu Huang
Nando Metzger Rodrigo Caye Daudt Konrad Schindler
Photogrammetry and Remote Sensing, ETH Zürich

Monocular depth estimation

Fine-tuned “Stable Diffusion” to model **P(depth I image)**

Training takes 2.5 days on a single NVIDIA RTX 4090 GPU

Trained on **Hypersim** (a photorealistic indoor scene dataset)

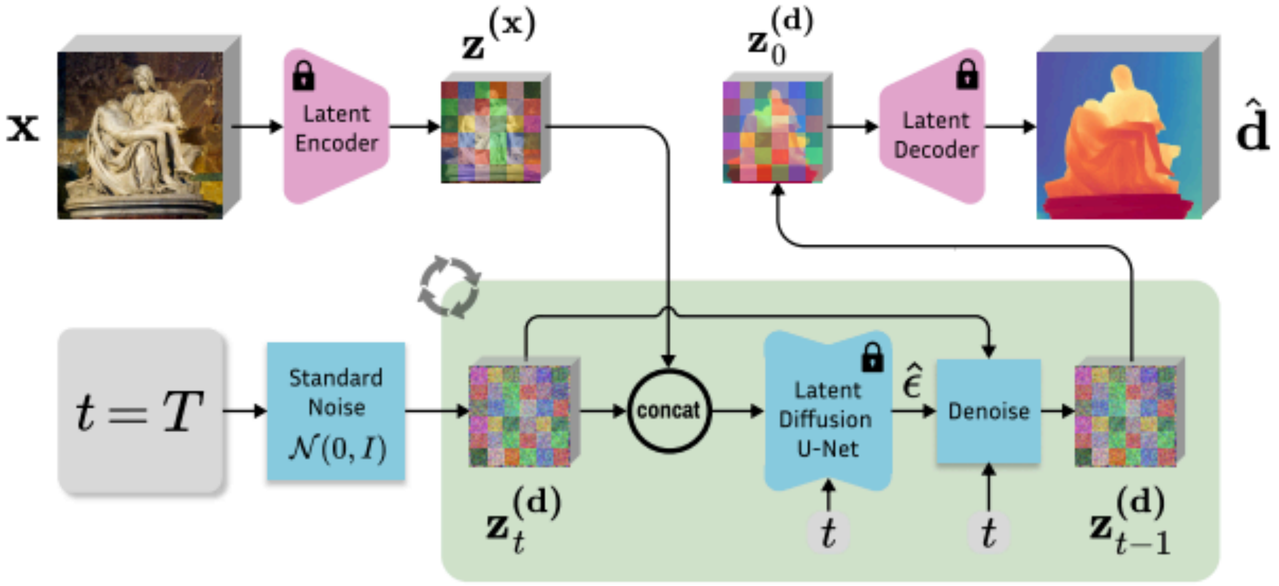
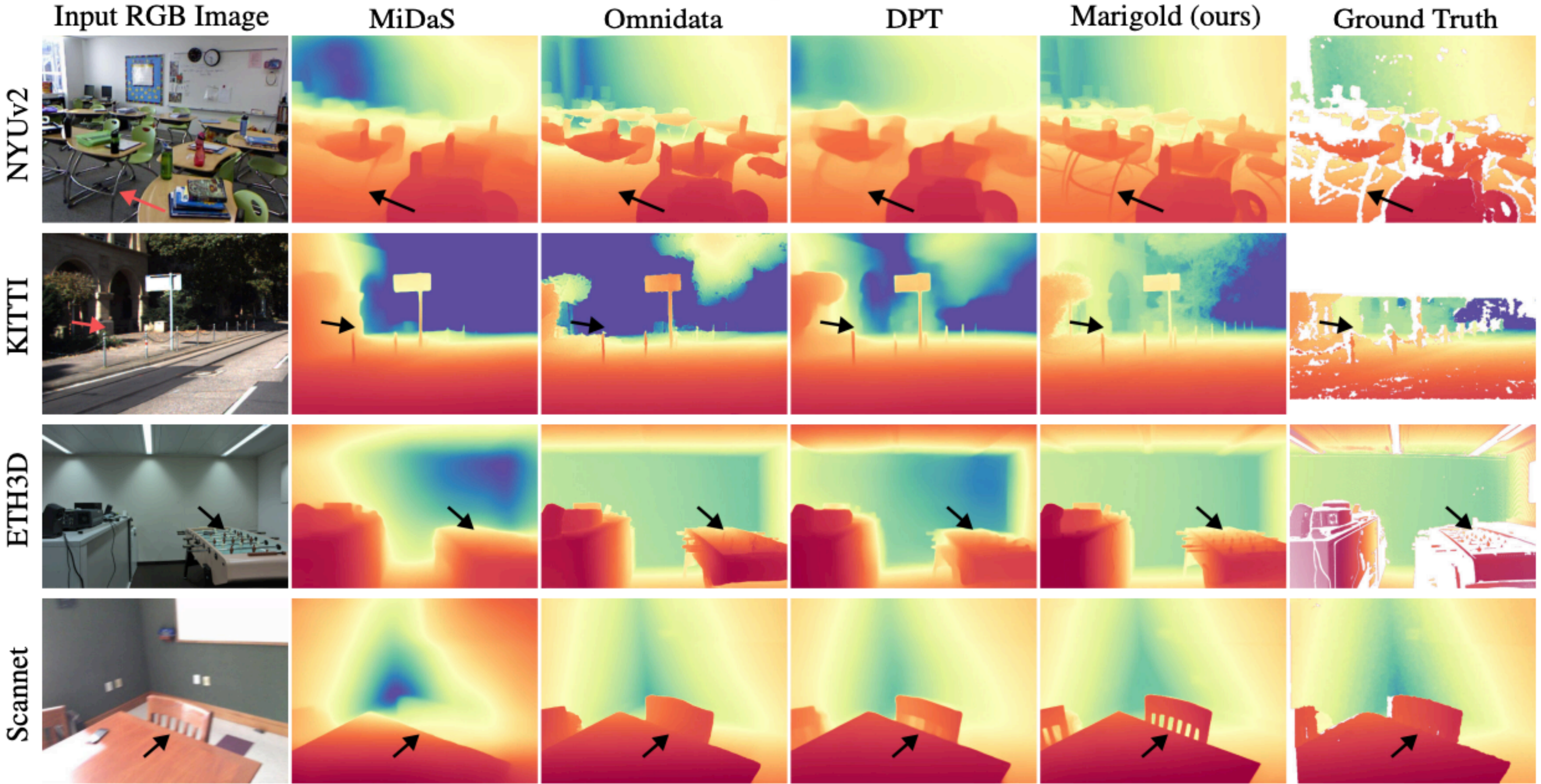


Figure 3. **Overview of the Marigold inference scheme.** Given an input image $\mathbf{x}$, we encode it with the original Stable Diffusion VAE into the latent code $\mathbf{z}^{(\mathbf{x})}$, and concatenate with the depth latent $\mathbf{z}_t^{(\mathbf{d})}$ before giving it to the modified fine-tuned U-Net on every denoising iteration. After executing the schedule of T steps, the resulting depth latent $\mathbf{z}_0^{(\mathbf{d})}$ is decoded into an image, whose 3 channels are averaged to get the final estimation $\hat{\mathbf{d}}$. See Sec. 3.4 for details.

Monocular depth estimation



DUSt3R: Geometric 3D Vision Made Easy

Shuzhe Wang*, Vincent Leroy†, Yohann Cabon†, Boris Chidlovskii† and Jerome Revaud†
*Aalto University
shuzhe.wang@aalto.fi
†Naver Labs Europe
firstname.lastname@naverlabs.com

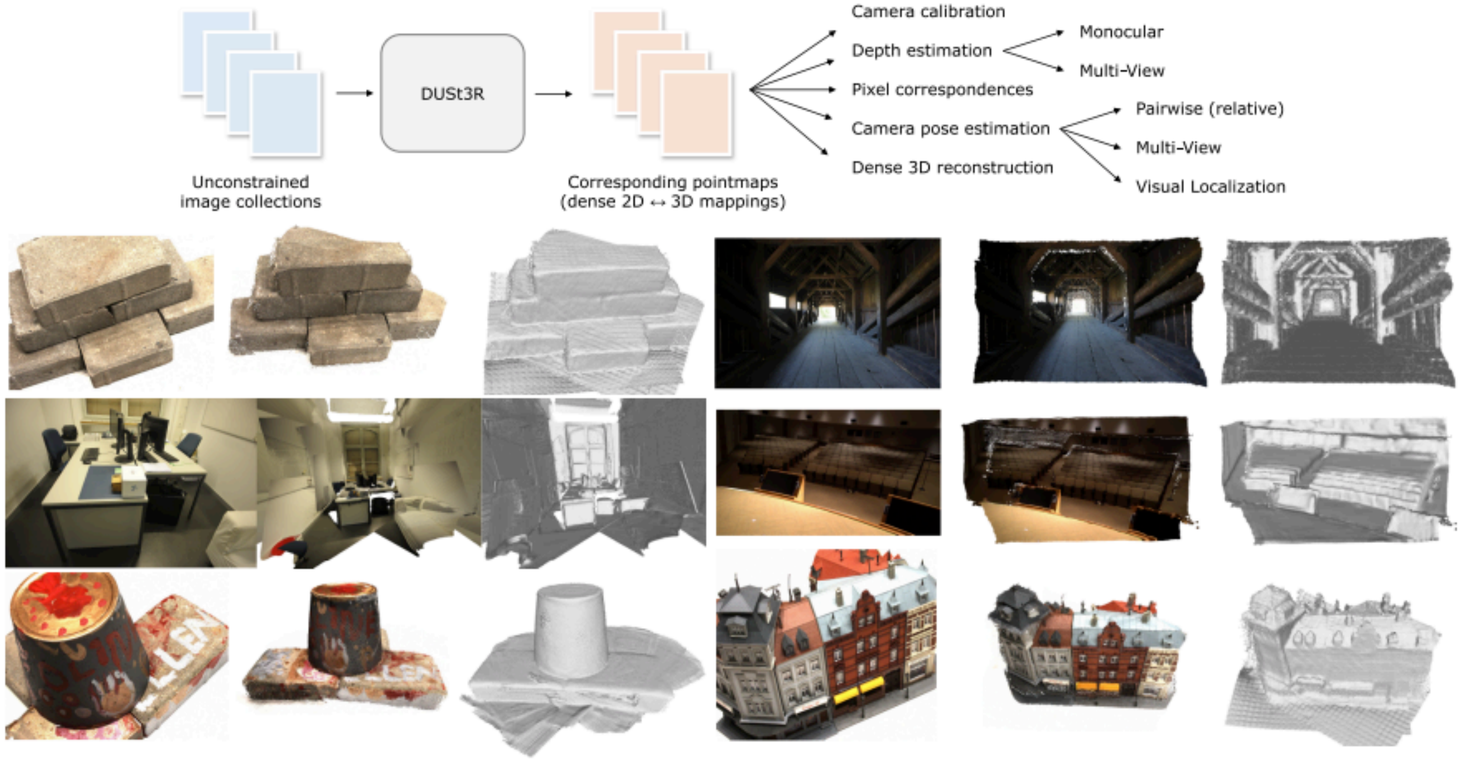


Figure 1. **Overview**: Given an unconstrained image collection, *i.e.* a set of photographs with unknown camera poses and intrinsics, our proposed method **DUSt3R** outputs a set of corresponding *pointmaps*, from which we can straightforwardly recover a variety of geometric quantities normally difficult to estimate all at once, such as the camera parameters, pixel correspondences, depthmaps, and fully-consistent 3D reconstruction. Note that DUSt3R also works for a single input image (*e.g.* achieving in this case monocular reconstruction). We also show **qualitative examples** on the DTU, Tanks and Temples and ETH-3D datasets [1, 51, 108] obtained **without** known camera parameters. For each sample, from *left* to *right*: input image, colored point cloud, and rendered with shading for a better view of the underlying geometry.

DUSt3R: Geometric 3D Vision Made Easy

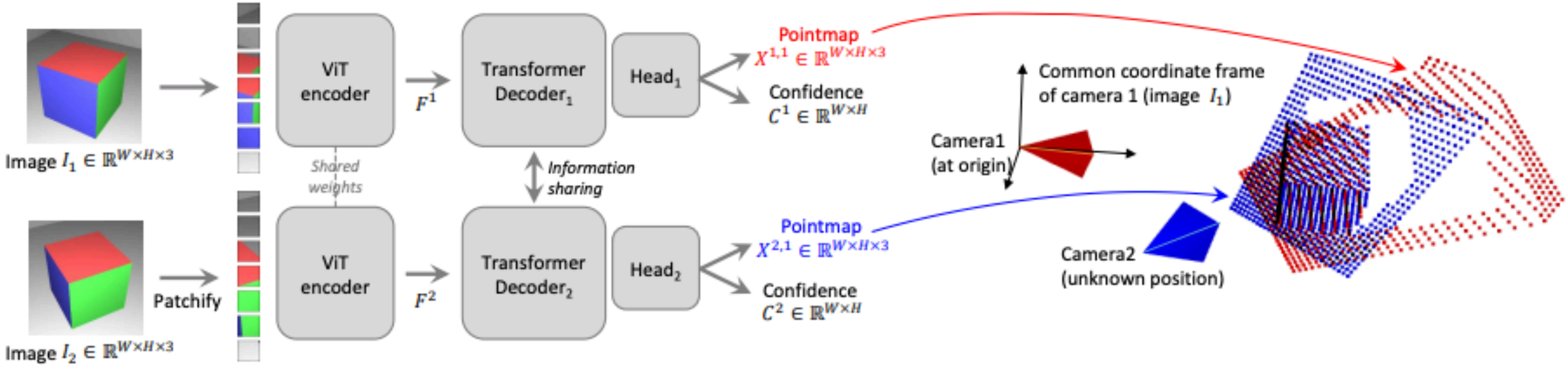
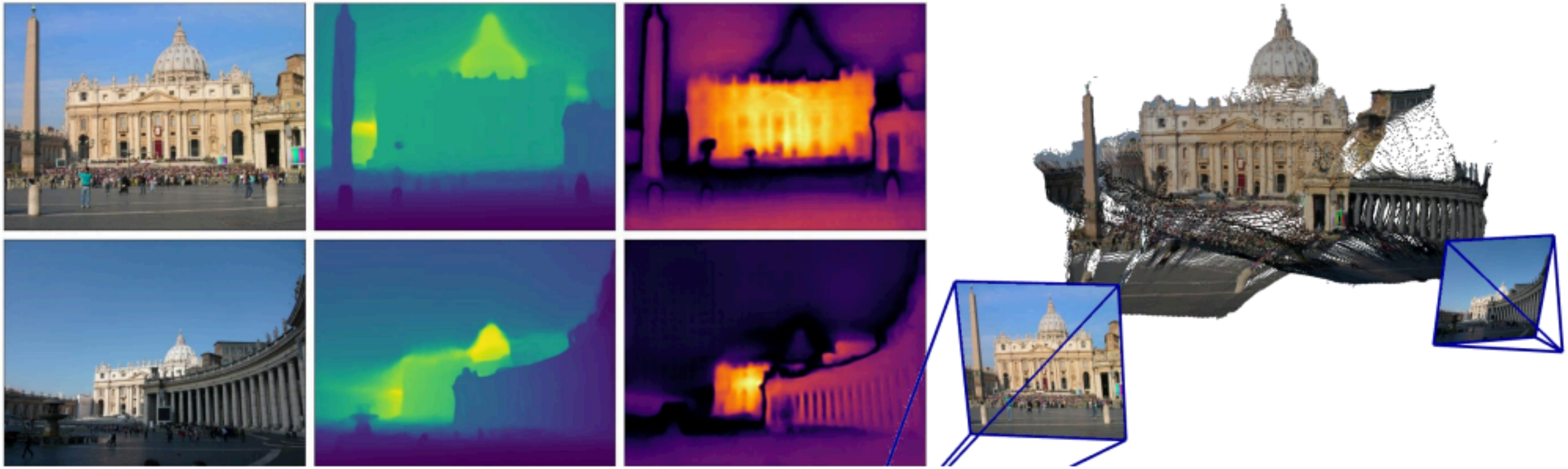
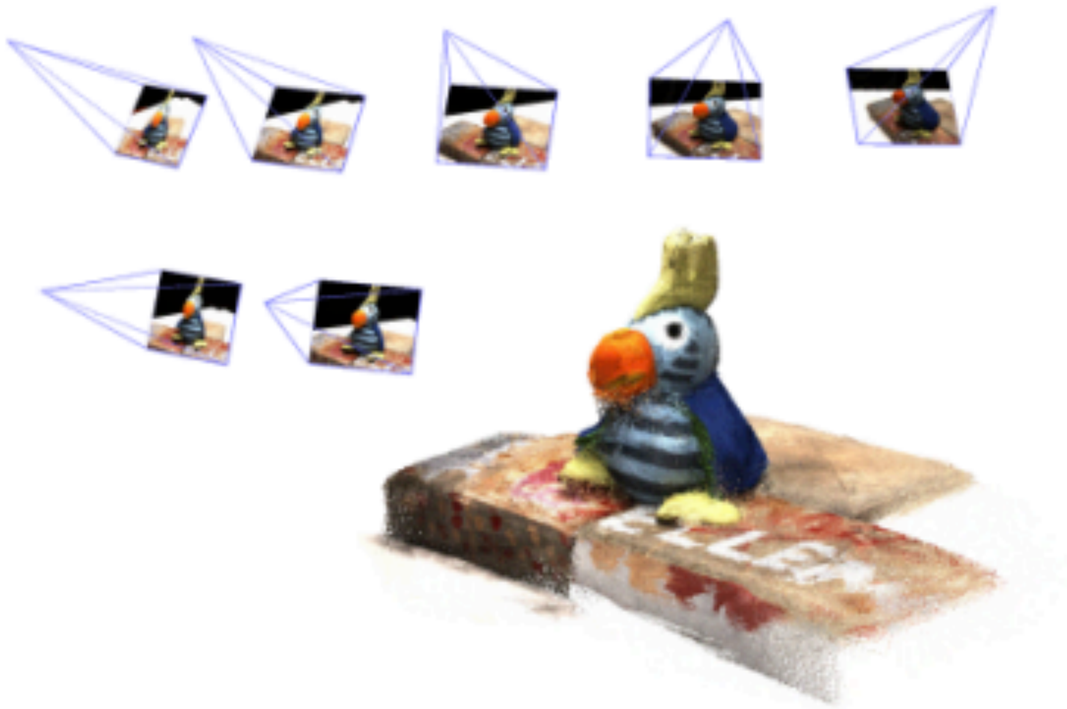


Figure 2. **Architecture of the network $\mathcal{F}$.** Two views of a scene (I^1, I^2) are first encoded in a Siamese manner with a shared ViT encoder. The resulting token representations F^1 and F^2 are then passed to two transformer decoders that constantly exchange information via cross-attention. Finally, two regression heads output the two corresponding pointmaps and associated confidence maps. Importantly, the two pointmaps are expressed in the same coordinate frame of the first image I^1. The network $\mathcal{F}$ is trained using a simple regression loss (Eq. (4))

DUSt3R: Geometric 3D Vision Made Easy



DUSt3R: Geometric 3D Vision Made Easy



VGGT: Visual Geometry Grounded Transformer

Jianyuan Wang[1,2] Minghao Chen[1,2] Nikita Karaev[1,2] Andrea Vedaldi[1,2]
Christian Rupprecht[1] David Novotny[2]

[1]Visual Geometry Group, University of Oxford [2]Meta AI

Figure 1. **VGGT** is a large feed-forward transformer with minimal 3D-inductive biases trained on a trove of 3D-annotated data. It accepts up to hundreds of images and predicts cameras, point maps, depth maps, and point tracks for all images at once in less than a second, which often outperforms optimization-based alternatives without further processing.

VGGT: Visual Geometry Grounded Transformer

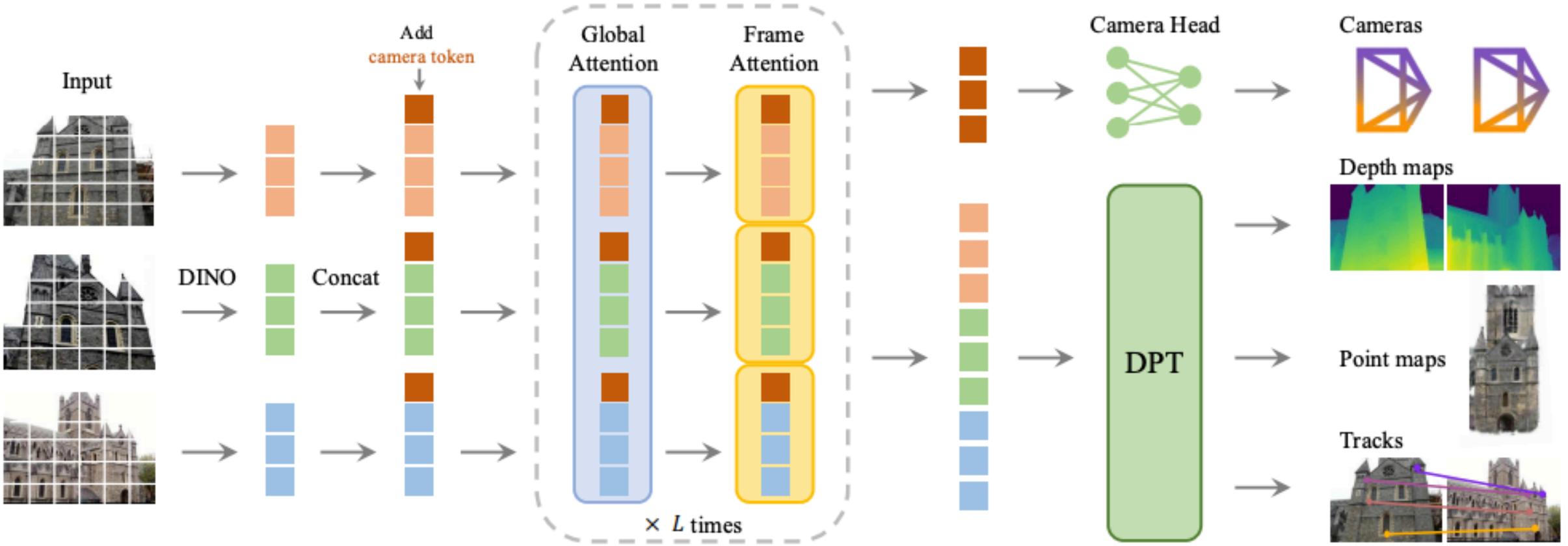


Figure 2. **Architecture Overview.** Our model first patchifies the input images into tokens by DINO, and appends camera tokens for camera prediction. It then alternates between frame-wise and global self attention layers. A camera head makes the final prediction for camera extrinsics and intrinsics, and a DPT [87] head for any dense output.

Depth Anything 3

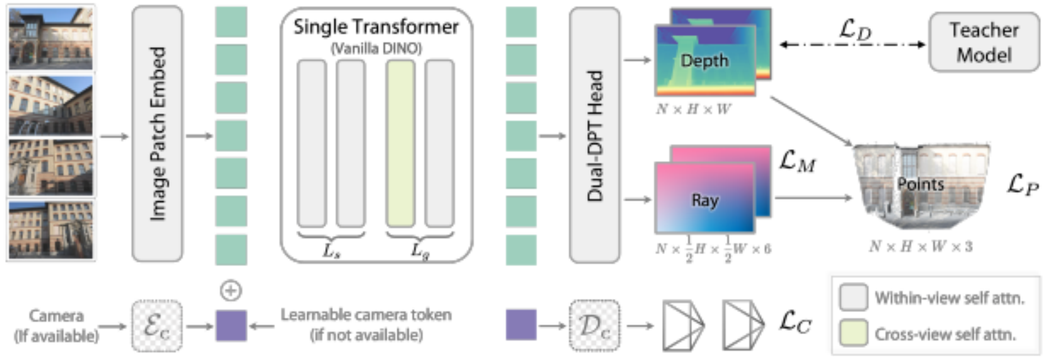


Figure 2 Pipeline of Depth Anything 3. Depth Anything 3 employs a single transformer (vanilla DINOv2 model) without any architectural modifications. To enable cross-view reasoning, an input-adaptive cross-view self-attention mechanism is introduced. A dual-DPT head is used to predict depth and ray maps from visual tokens. Camera parameters, if available, are encoded as camera tokens and concatenated with patch tokens, participating in all attention operations.

Depth Anything 3:
Recovering the Visual Space from Any Views

Haotong Lin*, **Sili Chen***, **Jun Hao Liew***, **Donny Y. Chen***, **Zhenyu Li**, **Guang Shi**, **Jiashi Feng**, **Bingyi Kang***,†

ByteDance Seed

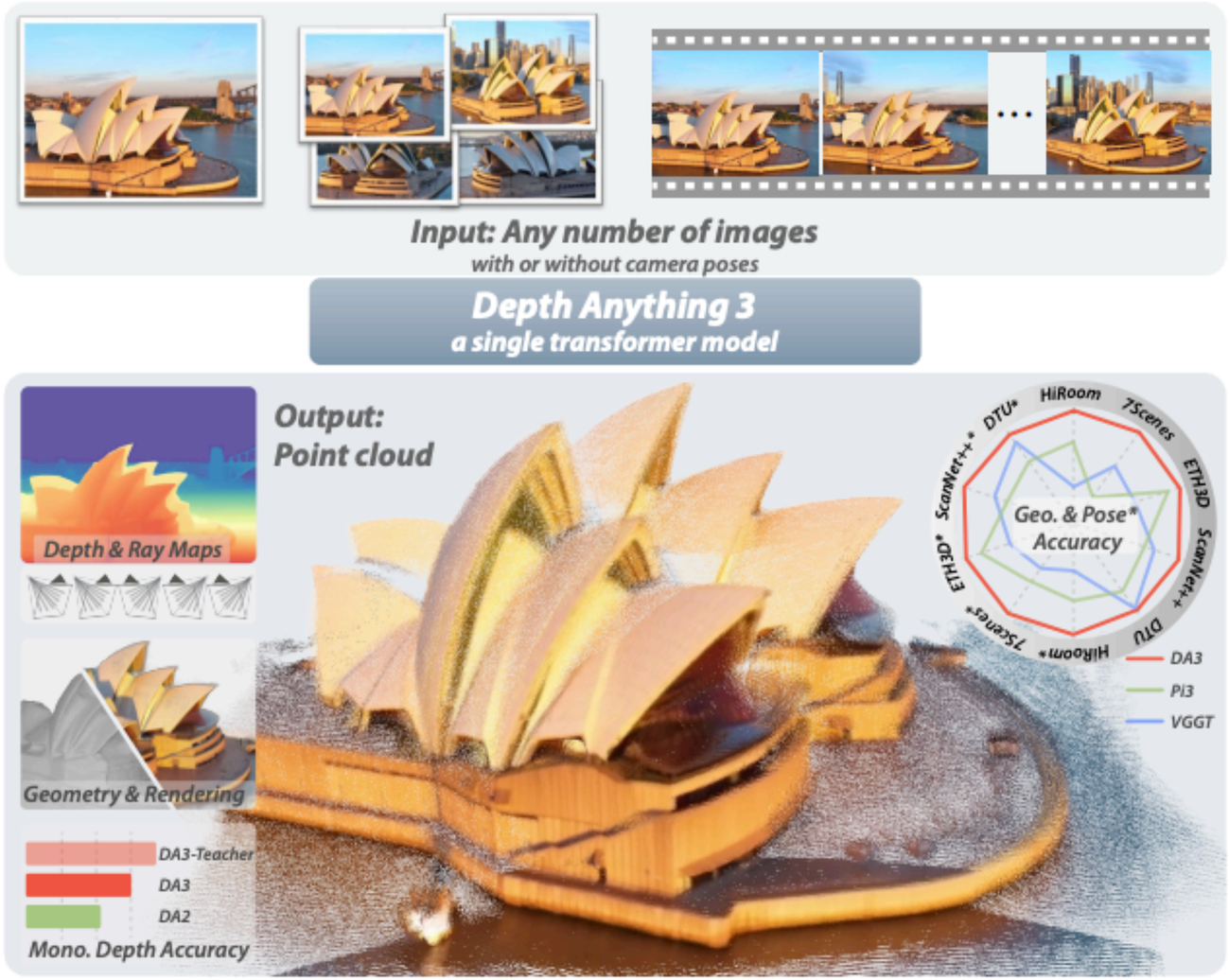


Figure 1 Given any number of images and optional camera poses, **Depth Anything 3** reconstructs the visual space, producing consistent depth and ray maps that can be fused into accurate point clouds, resulting in high-fidelity 3D Gaussians and geometry. It significantlyoutperforms VGGT in multi-view geometry and pose accuracy; with monocular inputs, it also surpasses Depth Anything 2 while matching its detail and robustness.

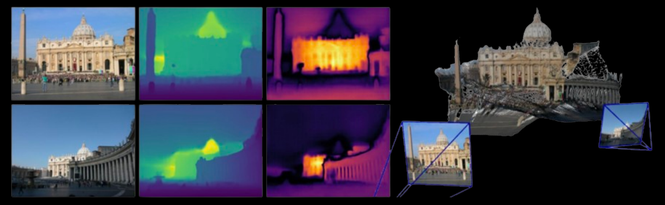
Summary

- Interests in **3D recognition models** are rapidly growing as 3D sensors and datasets become more accessible.

- Extending CNNs to 3D is non-trivial because 3D data like **polygon meshes** or **point clouds** lack regular grid structure.

- Recent advances in image to 3D combining **classical geometric pipelines** for **multi-view reconstruction**, **synthetic data**, **transformer architectures**.

- Many techniques based on **view-based methods, sparse convolutions, graph neural networks**, **transformers**, etc.



- However, **3D data** is still lacking relative to **images.**