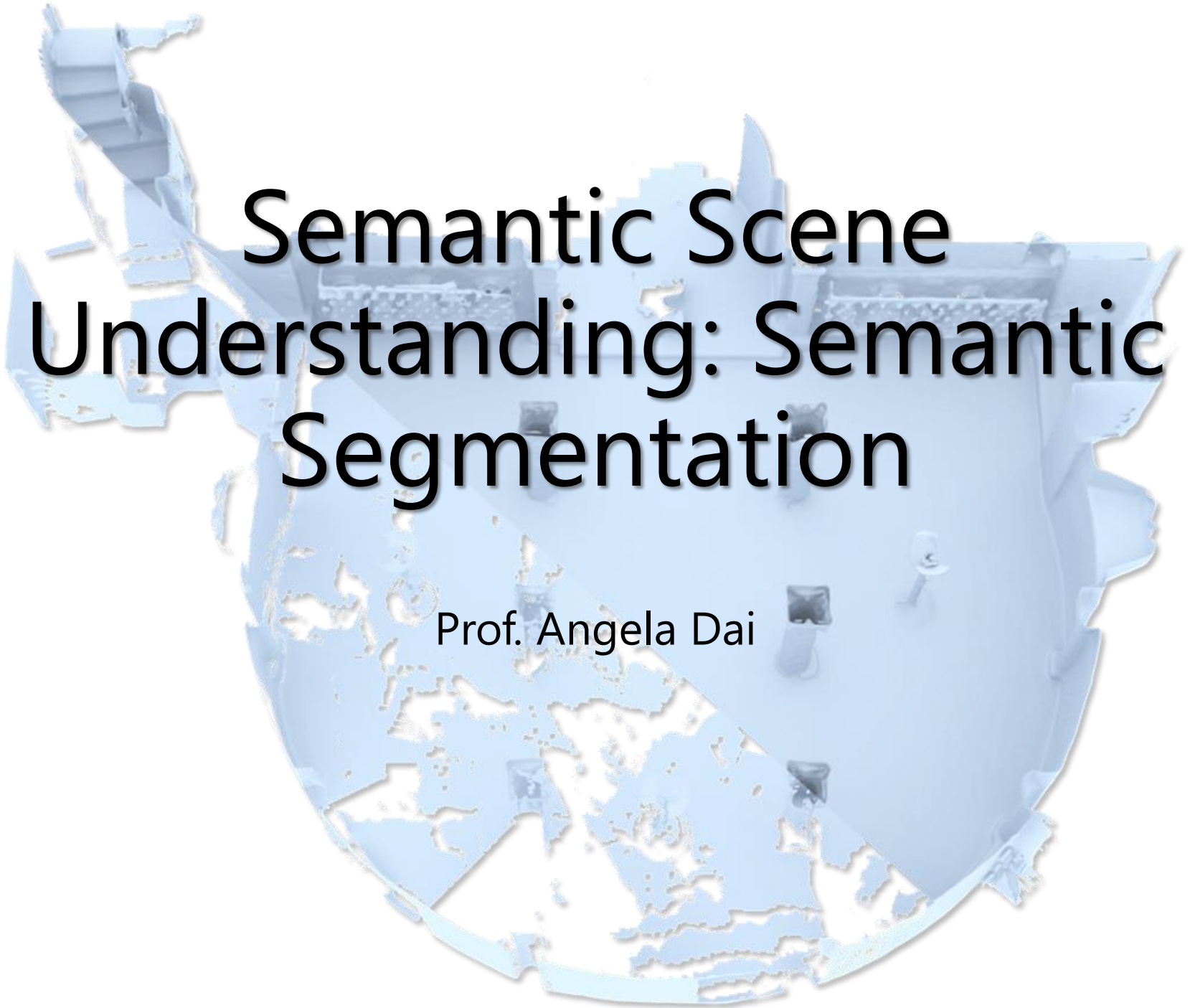
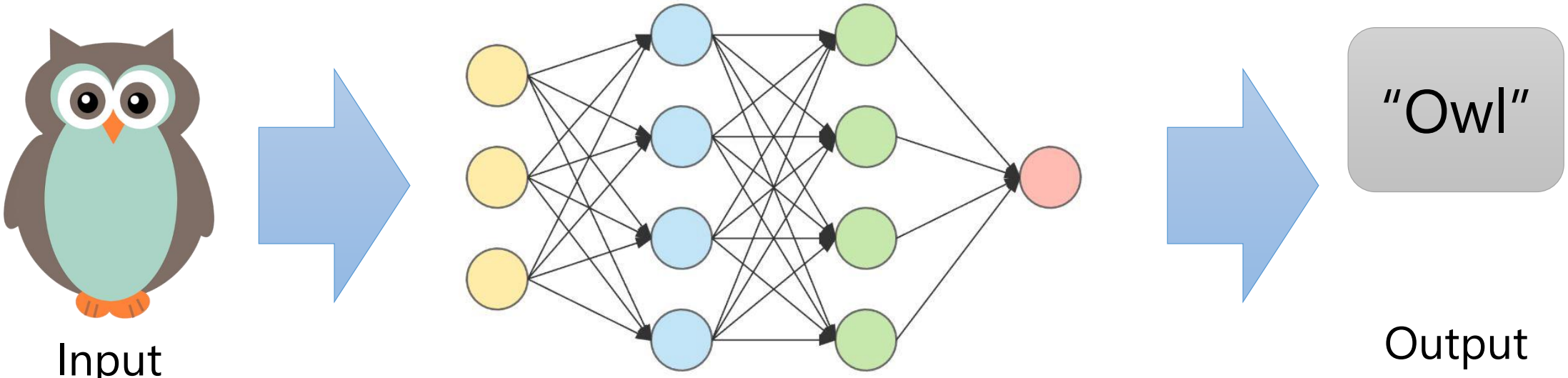


Semantic Scene Understanding: Semantic Segmentation

Prof. Angela Dai

Brief Recap

Deep Learning



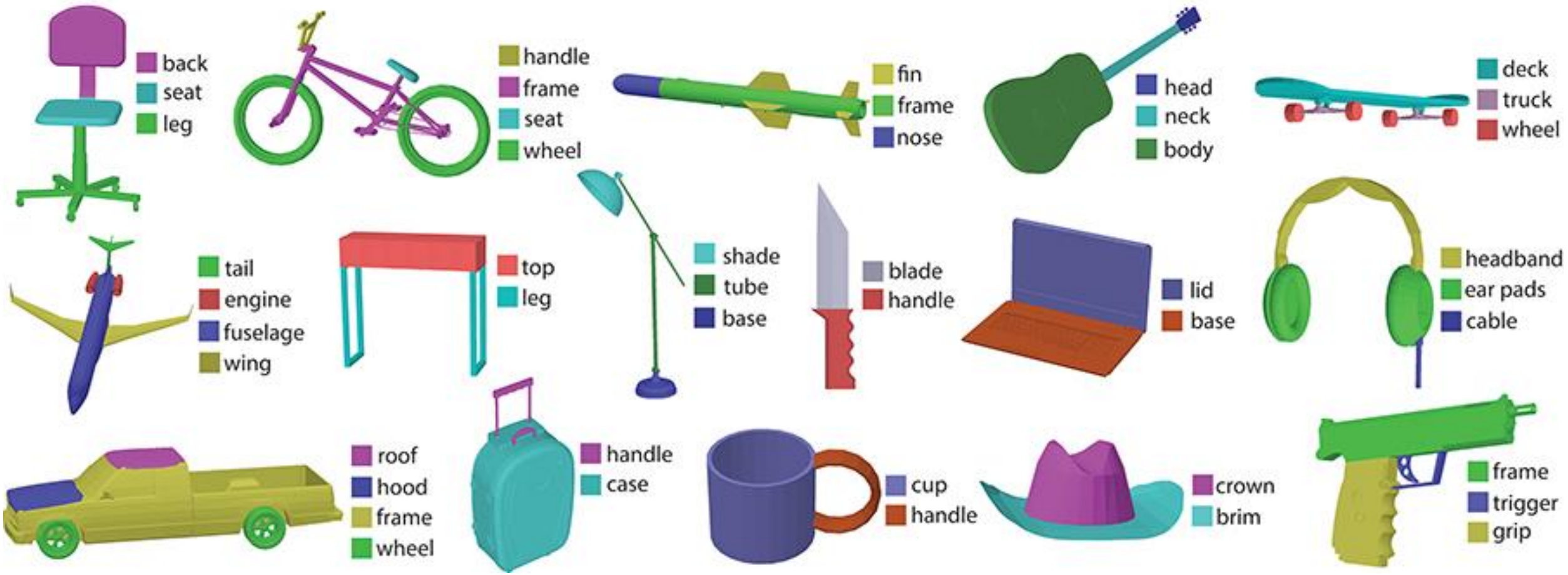
Input

Learned Feature Extraction
and Classification

Output

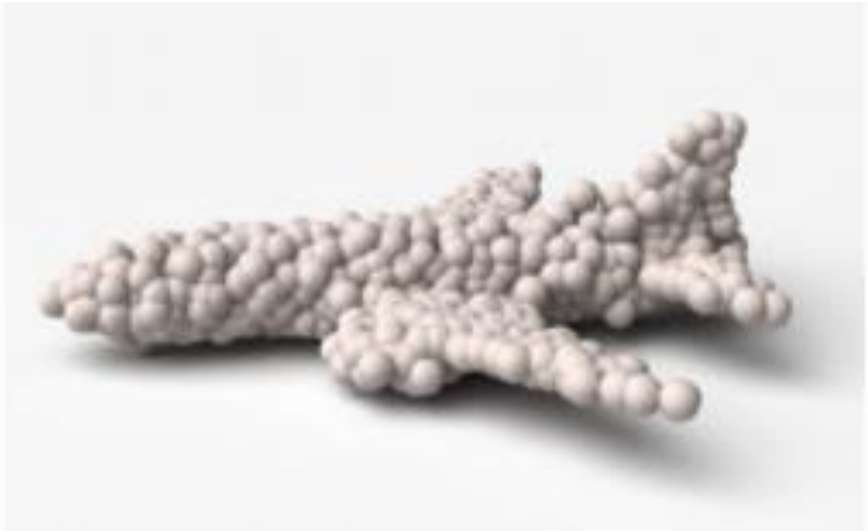
Want to automatically learn good feature representations for the task

Shape segmentation into parts

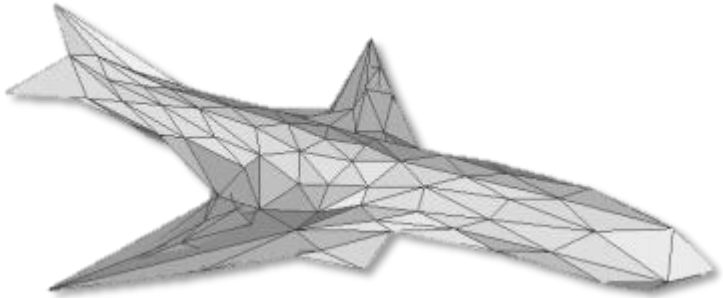


Generating Shapes

Signed Distance Fields

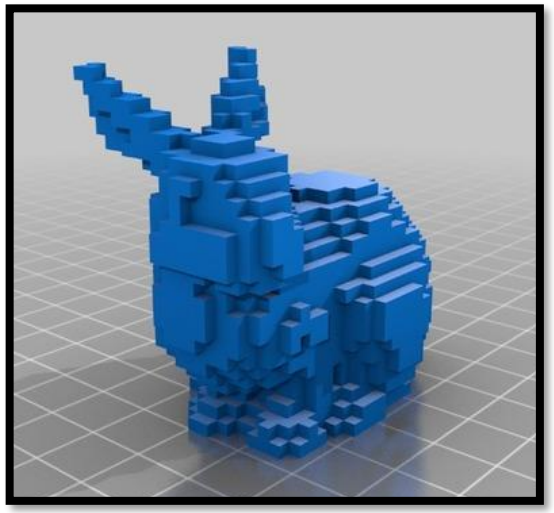


Point Clouds

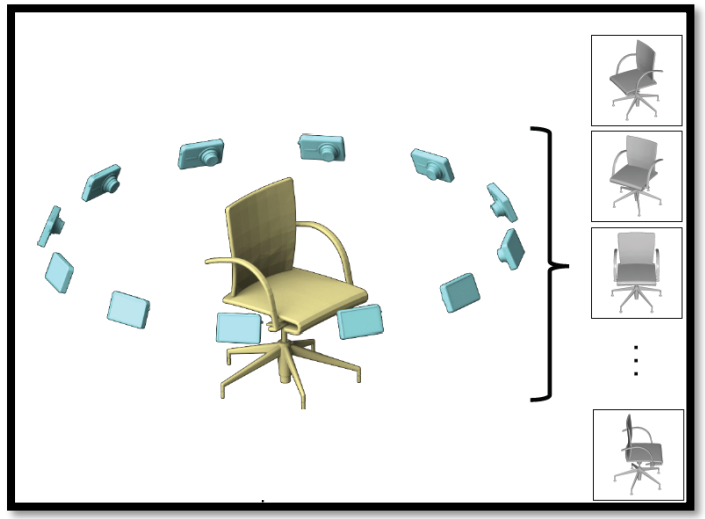


Meshes

3D Deep Learning by Representations

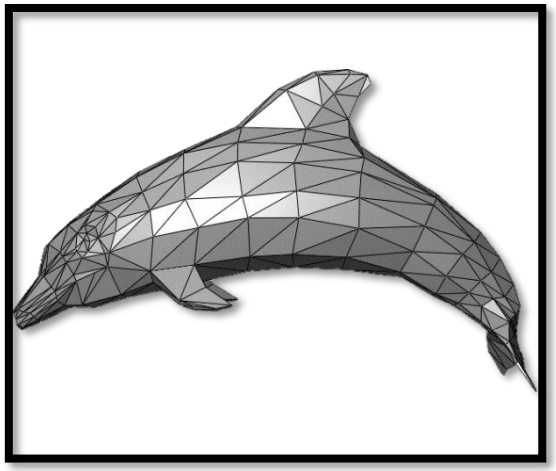


Volumetric
3D CNNs: Dense, Hierarchical, Sparse



Multi-View
(also: multi-view + volumetric/point/mesh)

Point Cloud



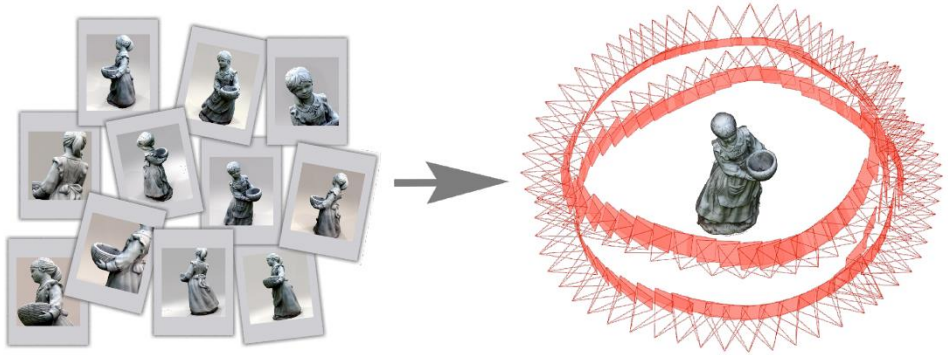
Mesh
Graph Neural Networks

and more!

Optimization & Learning

- Main difference (simplified): generalization
- Optimization: minimize loss on a train set
  - In computer science, a very general tool with broad applicability
- Machine learning: goal is generalization – minimizing loss on unseen samples

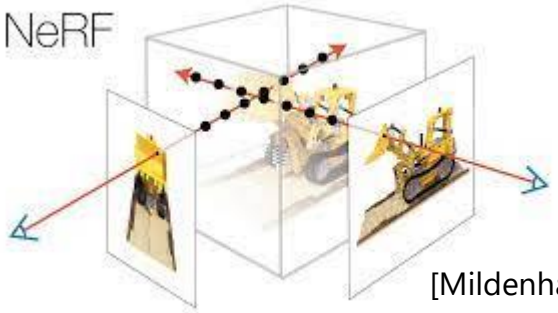
Example



Structure-from-Motion (3D Reconstruction)

[Mescheder et al. 19]

Learned Single-View 3D Reconstruction



[Mildenhall et al. 20]

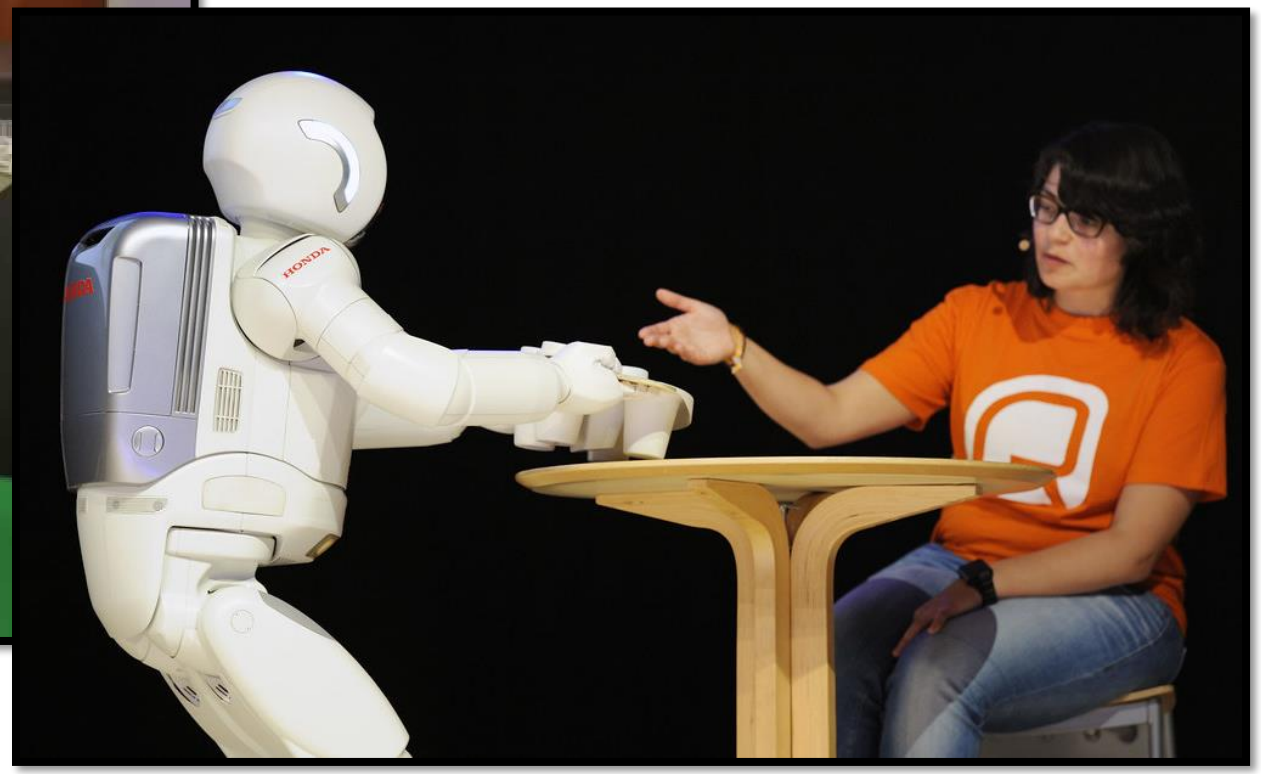
NeRF Optimization

[Yu et al. 21]

Generalized NeRF Prior for Few-Image Reconstruction

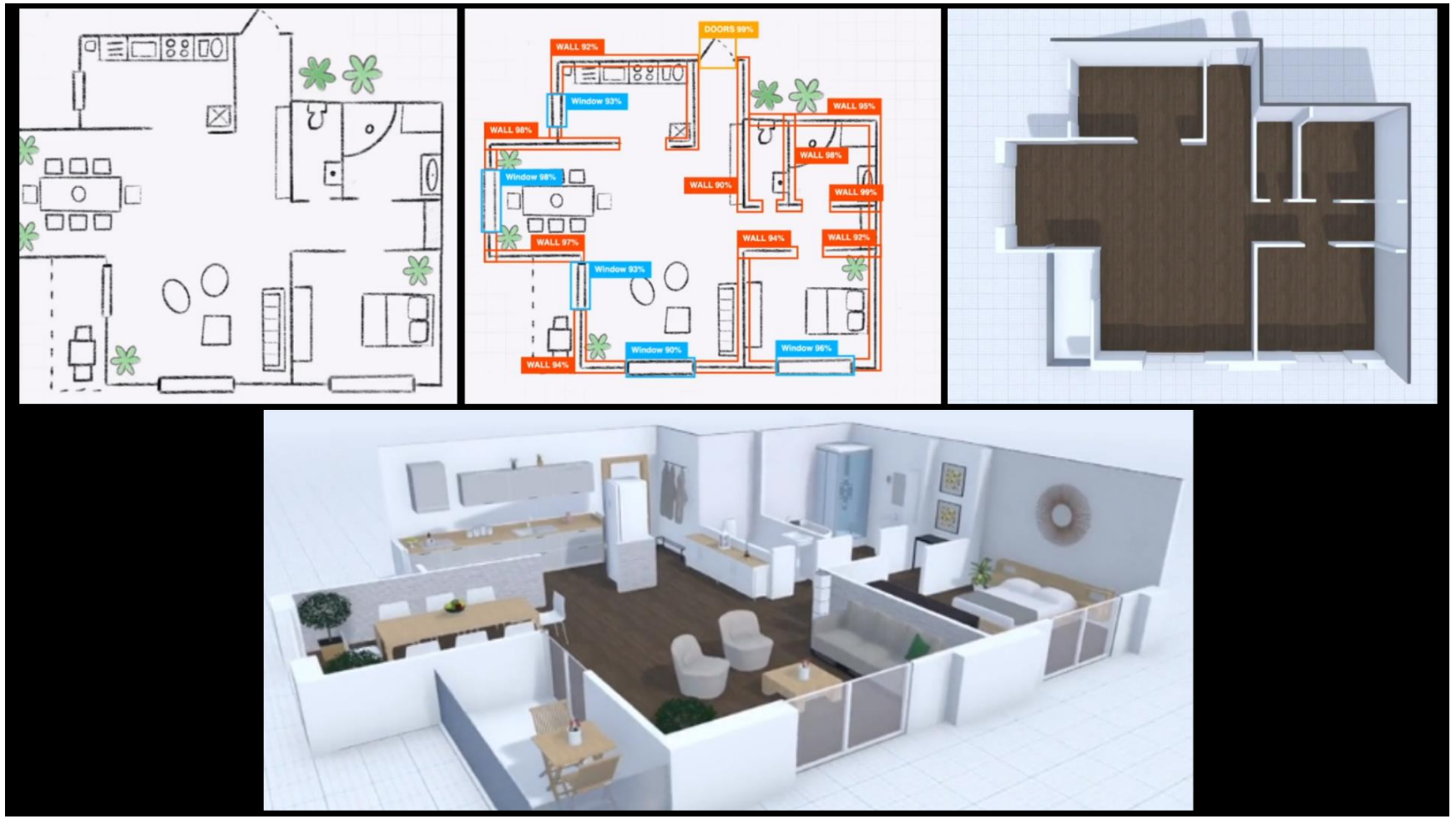
Understanding 3D Scenes

[Srivastava et al. ‘15]



ASIMO

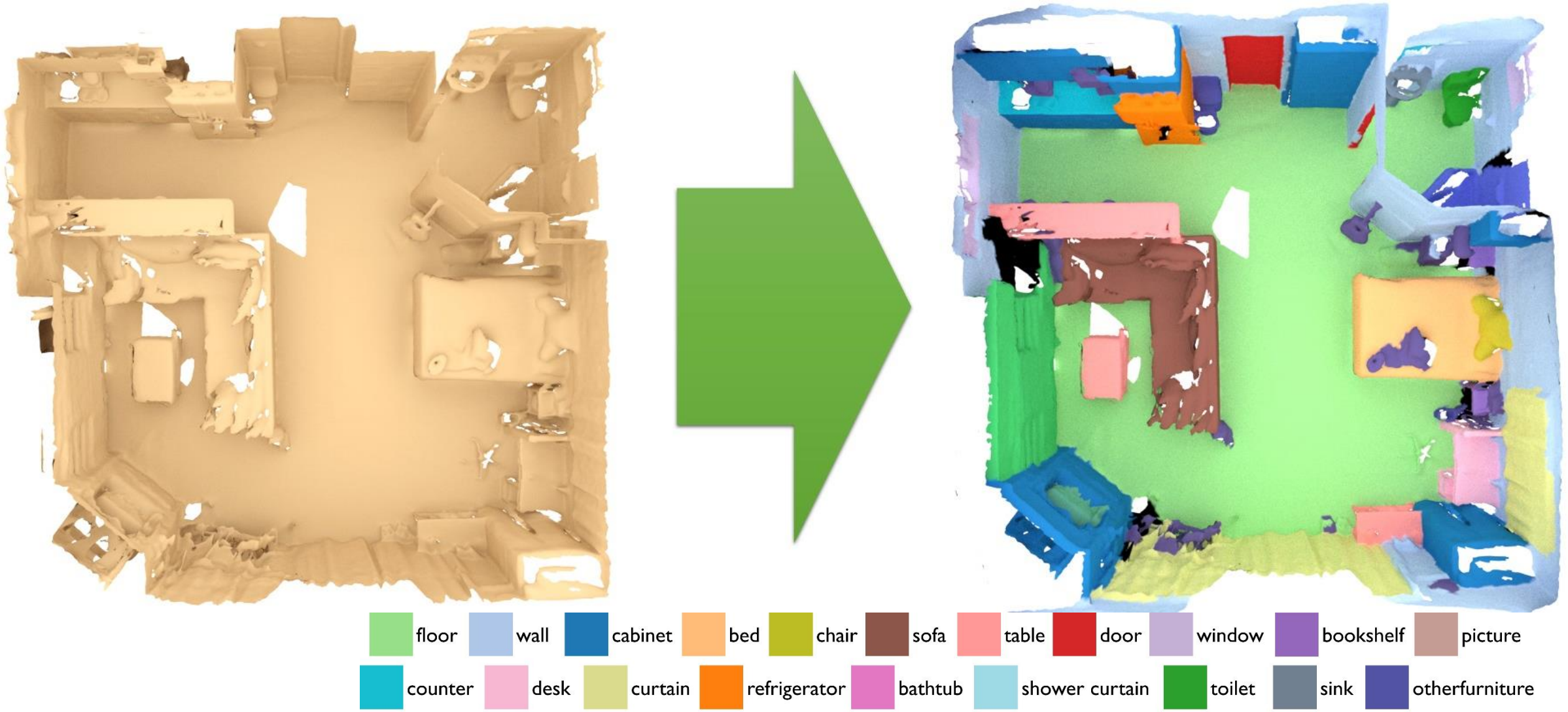
Understanding 3D Scenes



Understanding 3D Scenes

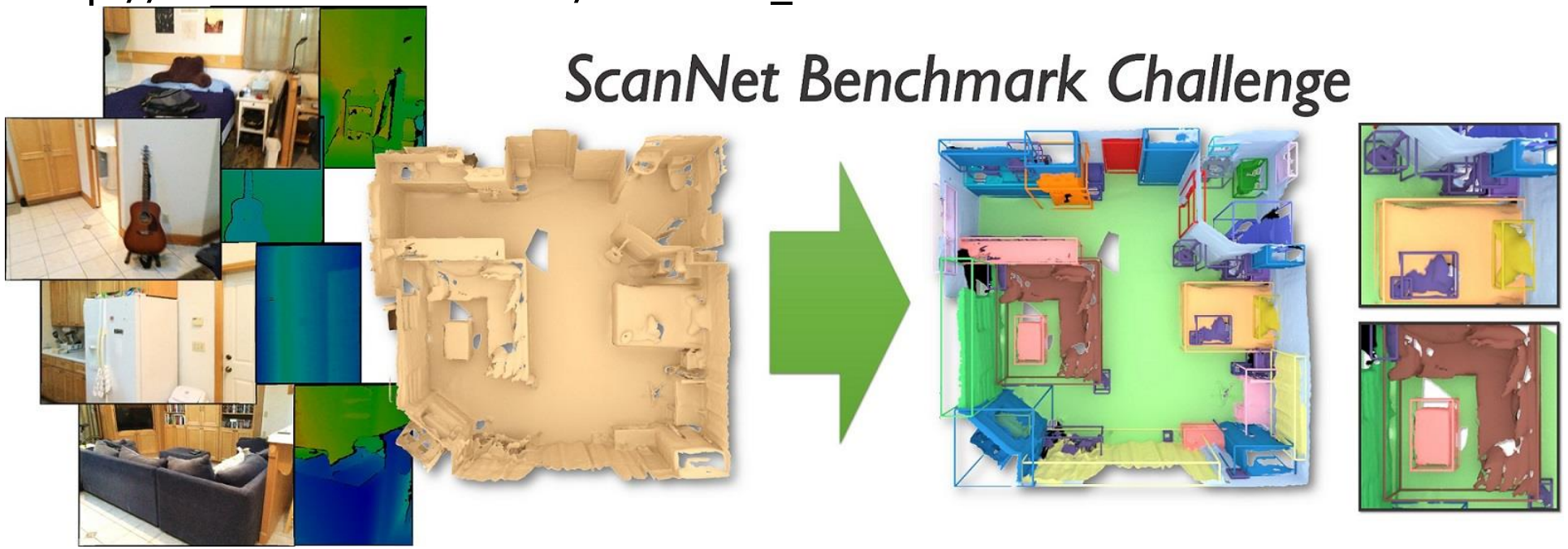
Understanding 3D Scenes

3D Semantic Segmentation



3D Semantic Segmentation

- Popular benchmarks:
- ScanNet Benchmark:
  - http://kaldir.vc.in.tum.de/scannet_benchmark

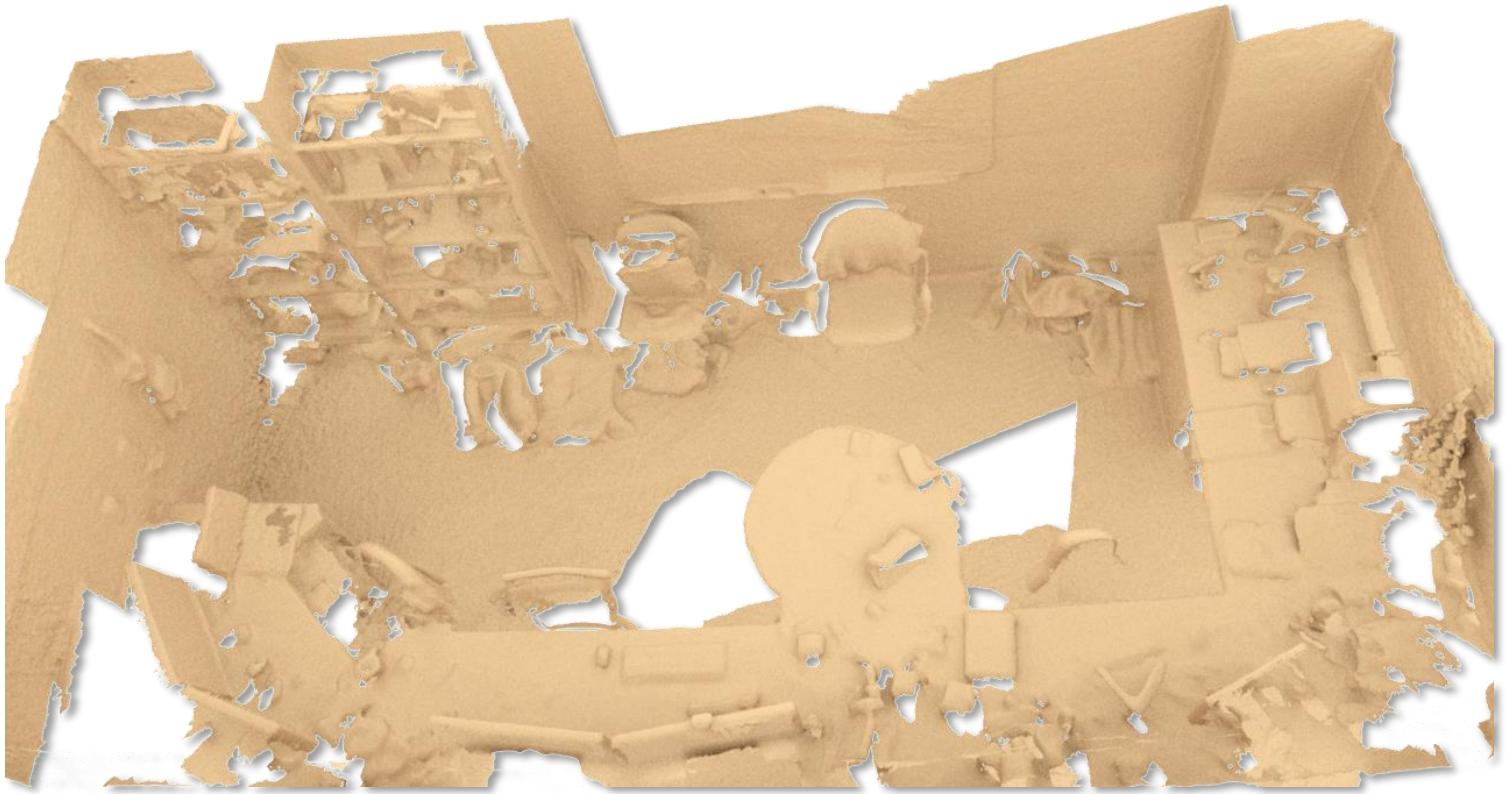


3D Semantic Segmentation

- Popular datasets:
- KITTI / KITTI-360:
  - http://www.cvlibs.net/datasets/kitti

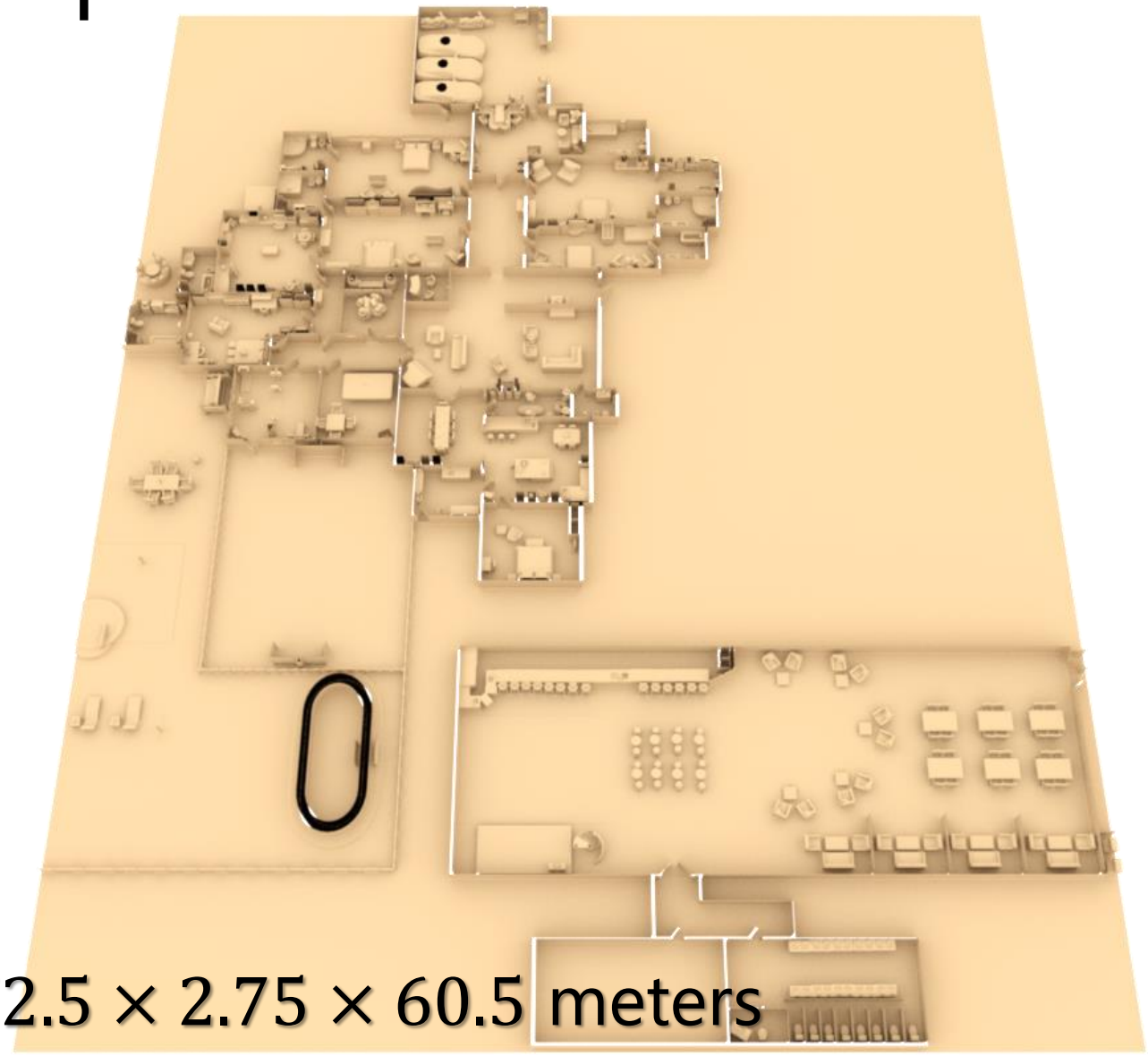
Related: Part Segmentation

Shapes vs. Scenes



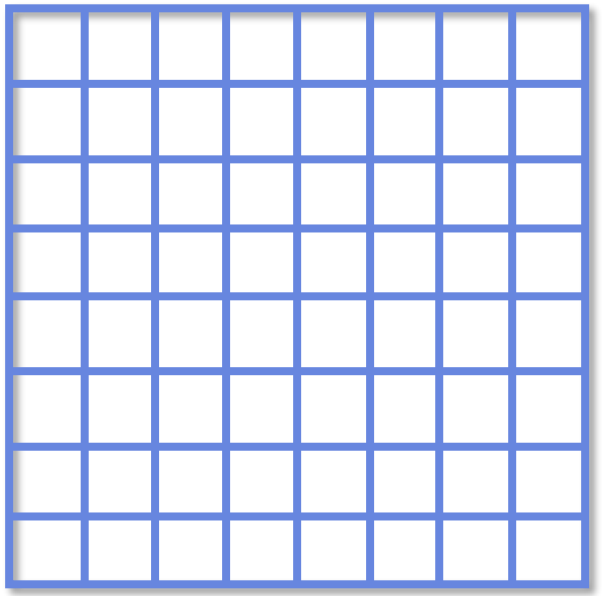
3D Scenes: Varying Spatial Extents

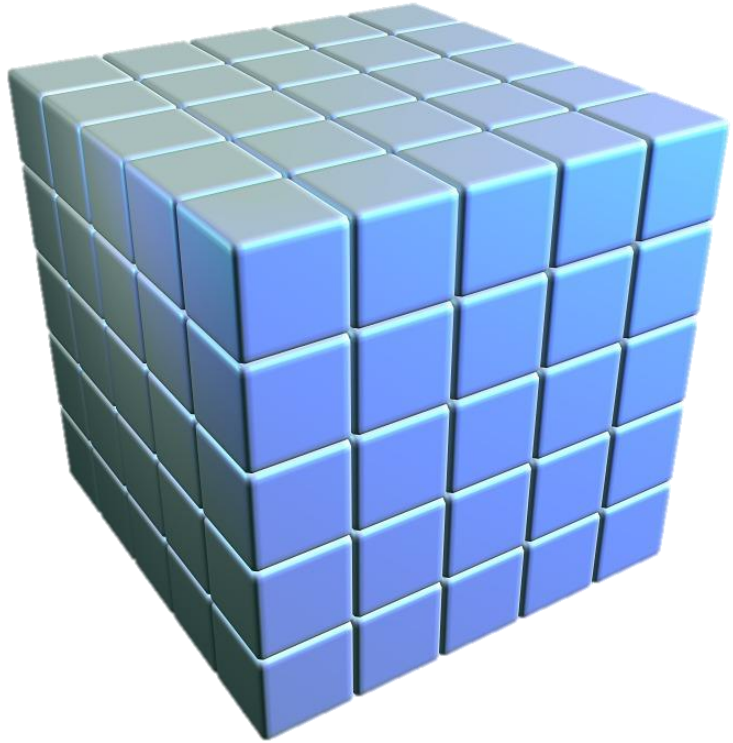
6 × 2.75 × 5.2 meters

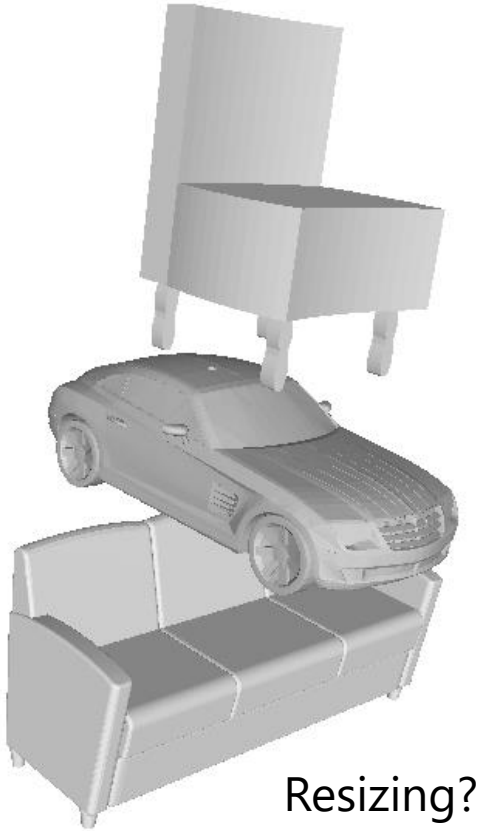


72.5 × 2.75 × 60.5 meters

3D Scenes vs 2D Images



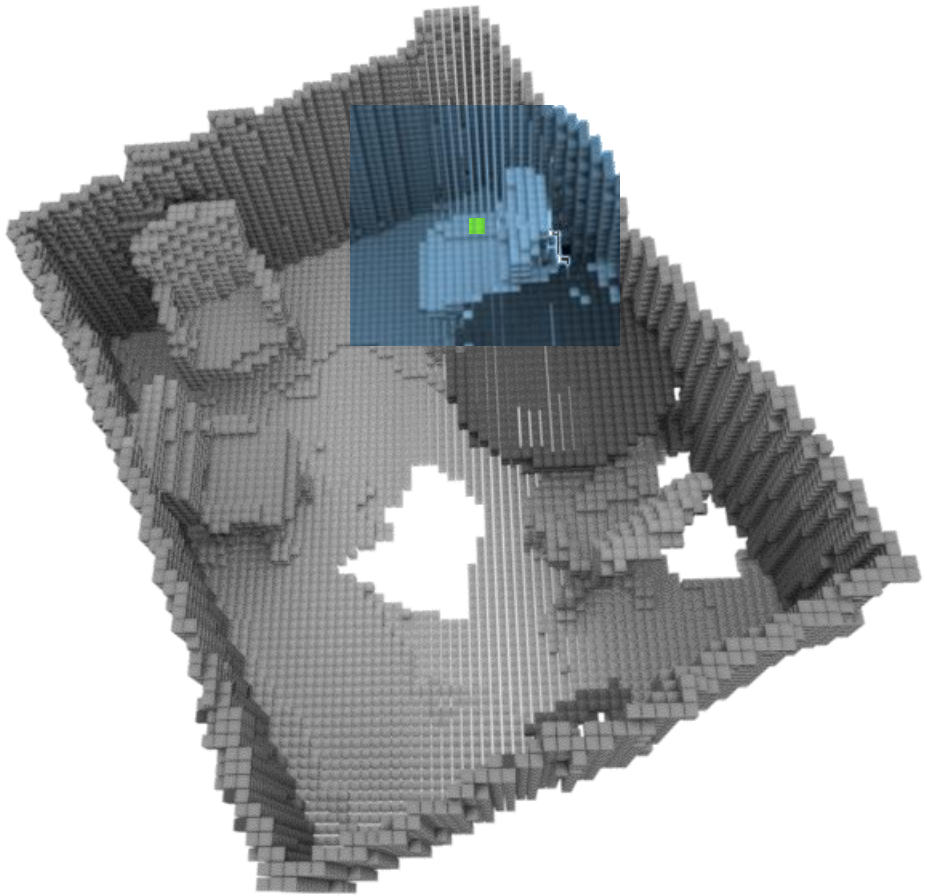




Resizing?

Original ScanNet Semantic Segmentation

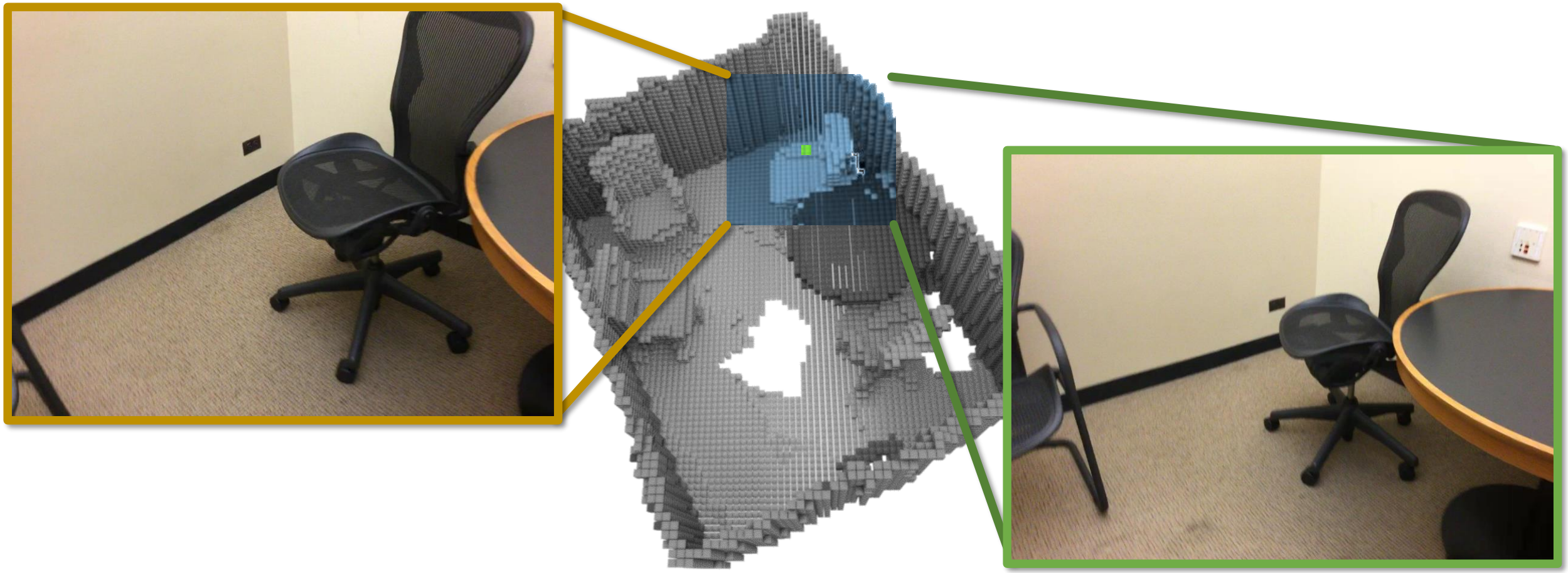
Predicting column by column



Use local neighborhood context

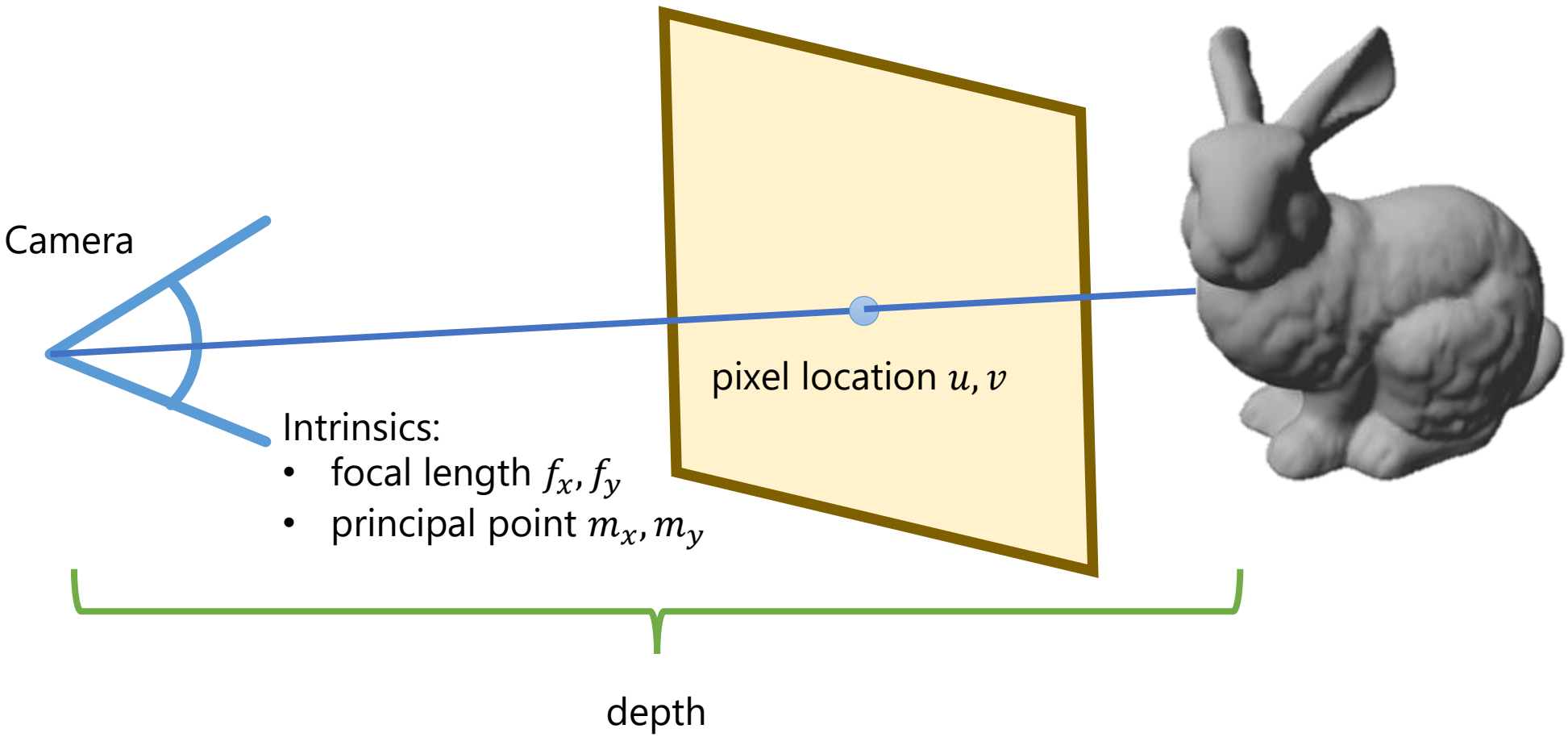
[Dai et al. '17]

Joint 3D and Multi-View



[Dai et al. ‘18]

Projection between 3D and 2D



Projection between 3D and 2D

- Point in 3D: (x, y, z)
- Use camera intrinsics:

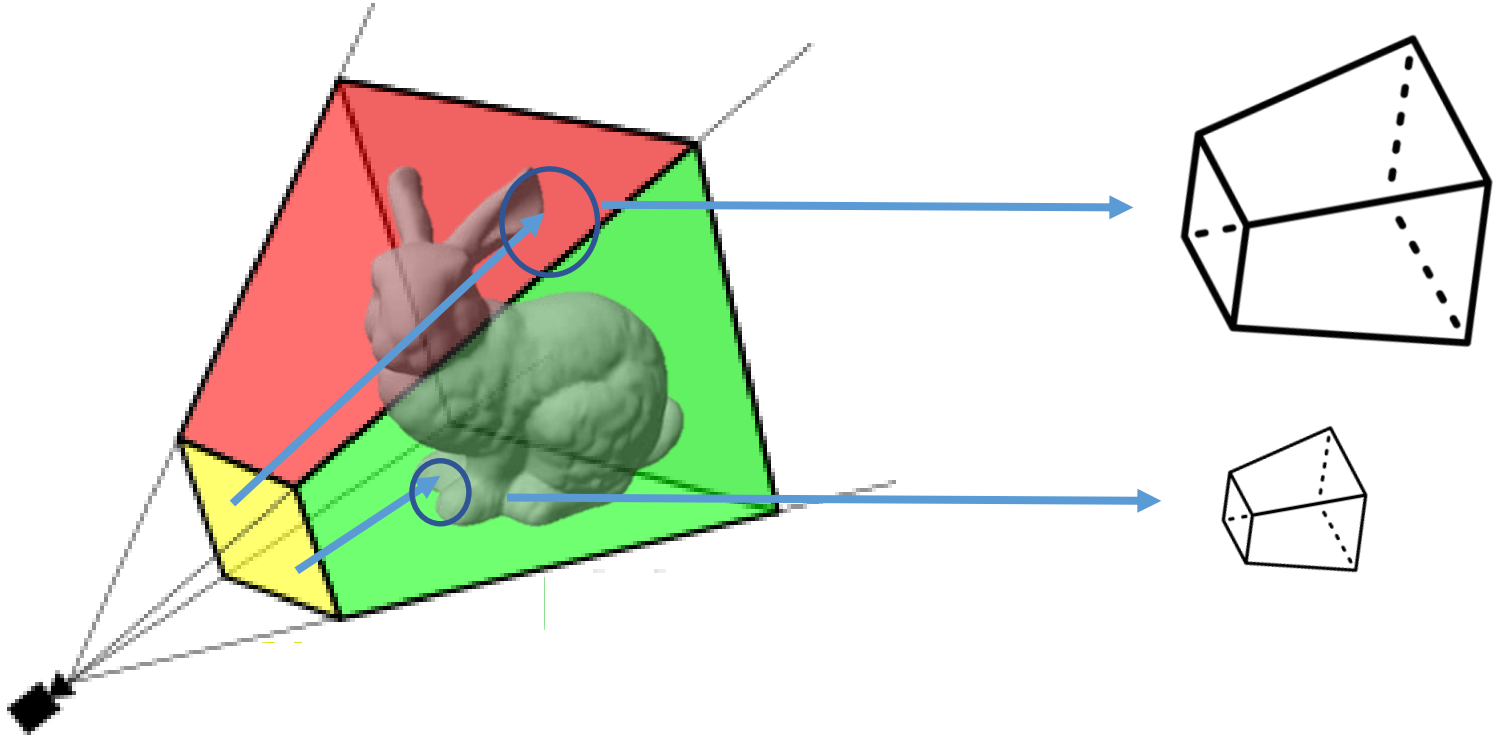
$$\begin{pmatrix} f_x & 0 & m_x \\ 0 & f_y & m_y \\ 0 & 0 & 1 \end{pmatrix}$$

- Projection:

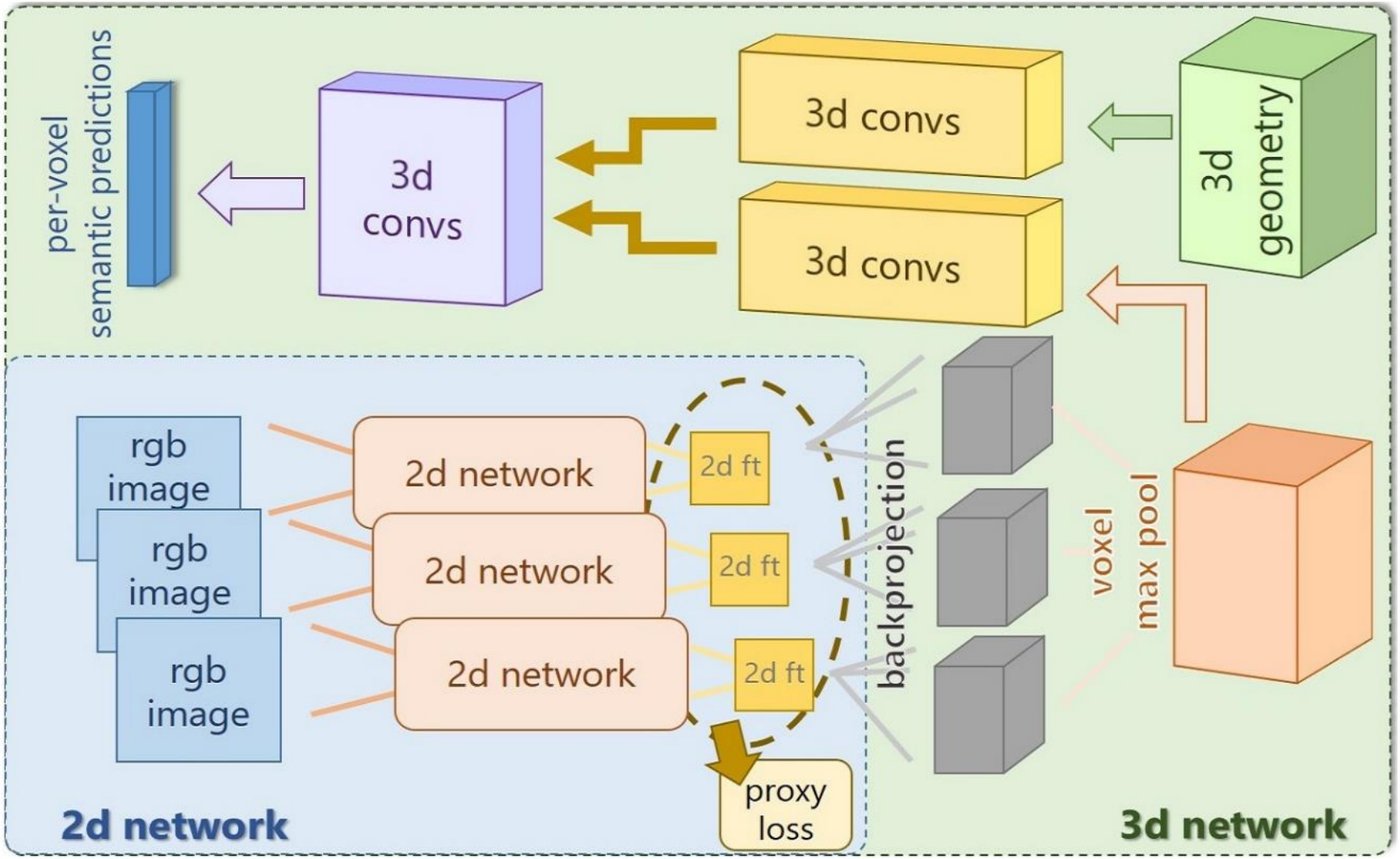
$$\begin{pmatrix} f_x & 0 & m_x \\ 0 & f_y & m_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

Projection between 3D and 2D

- Note: coverage of a 2D pixel can correspond to different 3D spatial extents

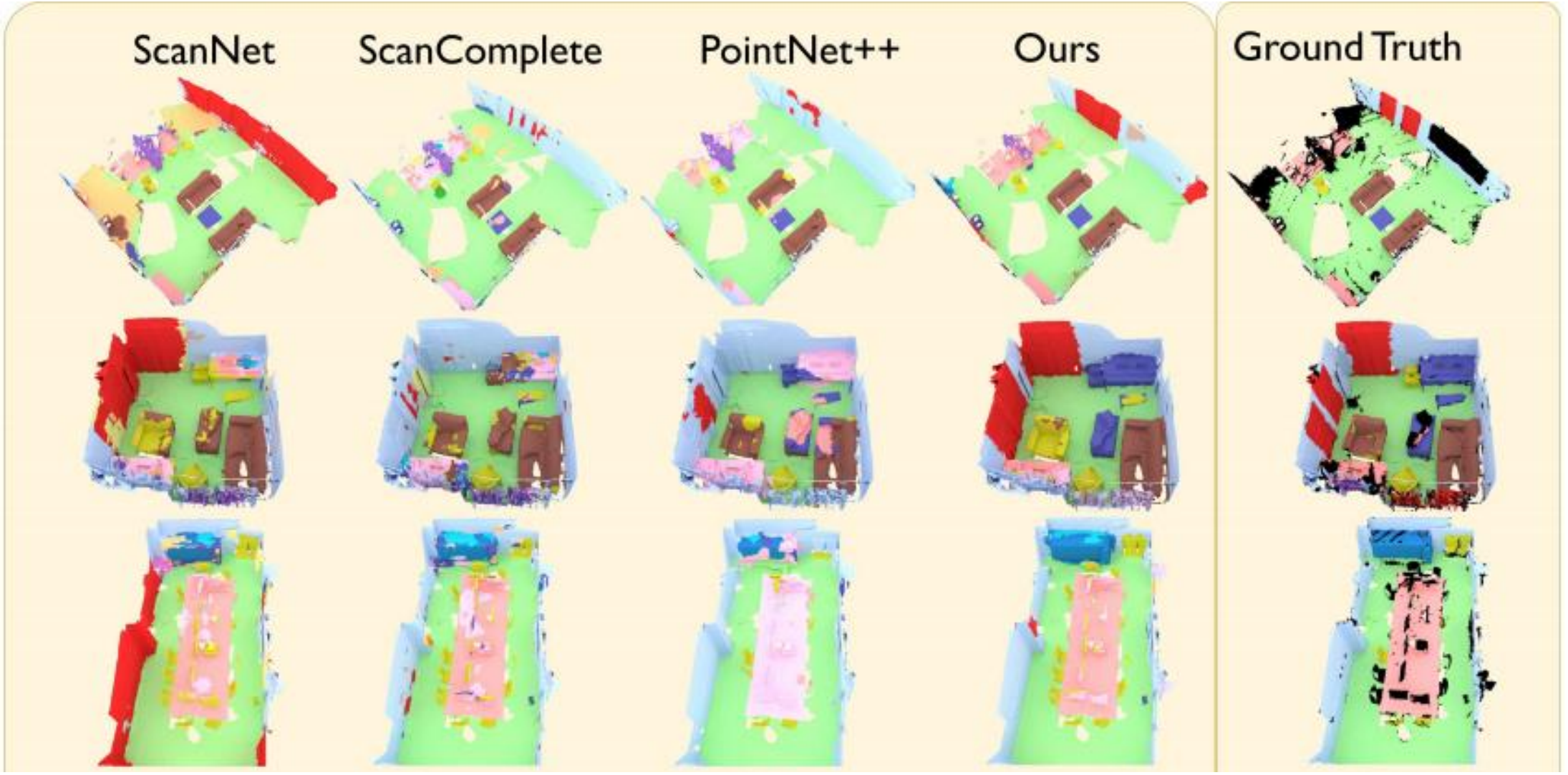


Joint 3D and Multi-View



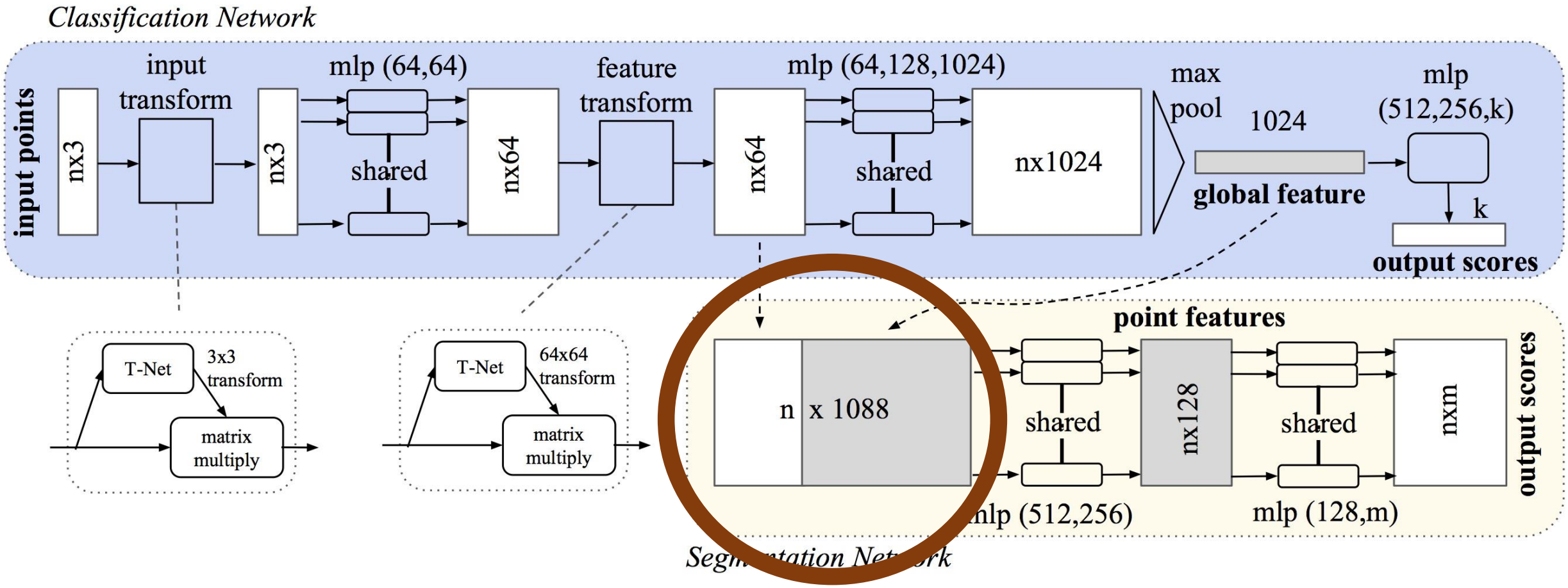
[Dai et al. '18]

Joint 3D and Multi-View



[Dai et al. '18]

PointNet

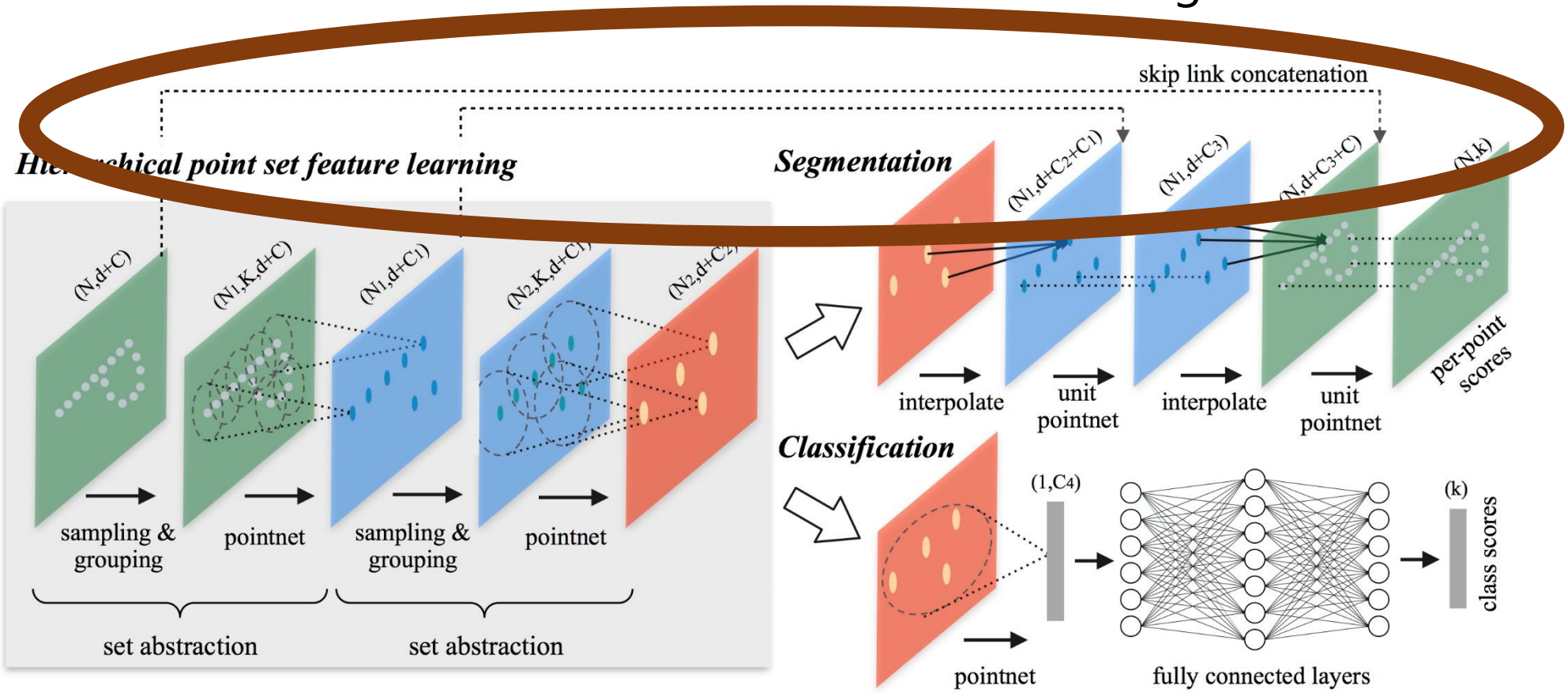


Concatenate local and global features

[Qi et al. '17]

PointNet++

Concatenate local and global features



[Qi et al. '17]

PointNet, PointNet++

- Chunk-by-chunk

[Qi et al. ‘17; Qi et al. ‘17]

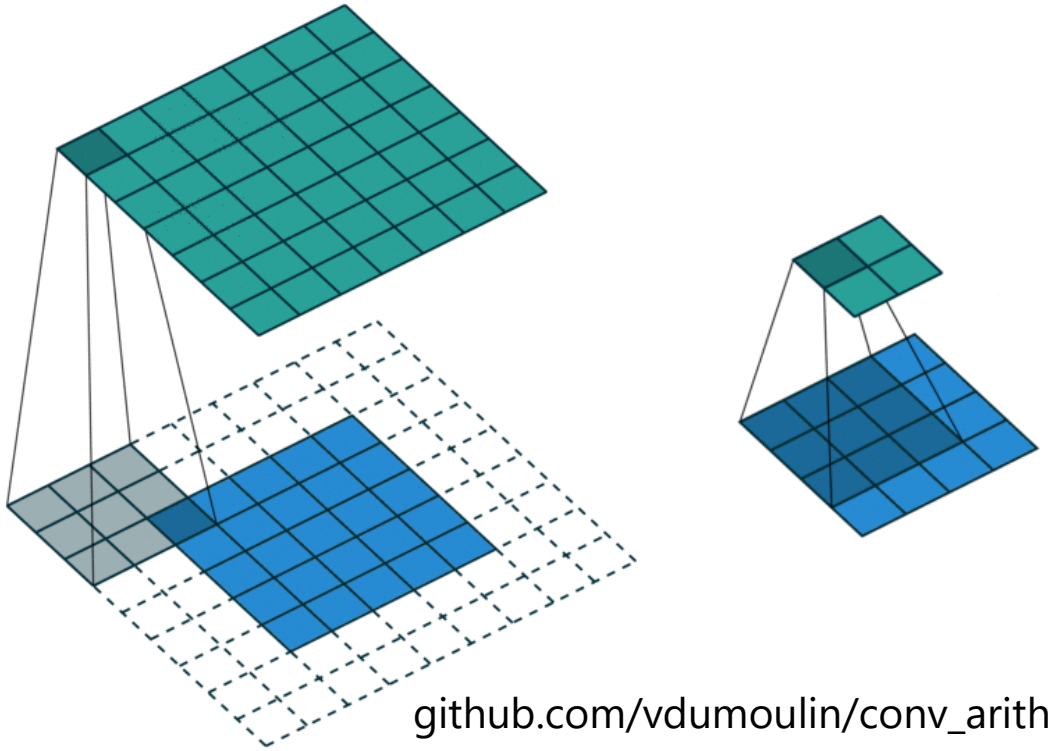
Sliding Window Processing of Scenes

- Adaptive to varying scene sizes
- For a $w \times h \times d$ scene, need to run $O(w \times h)$ times for inference
  - Subsample -> can have more inconsistencies
  - Voxel-by-voxel -> slow
- Limited context / receptive field by window size

ScanComplete: Fully-convolutional

- Fully-convolutional network architecture
- Recall: convolutions share weights

Same 3×3 kernel can be applied to various sized inputs

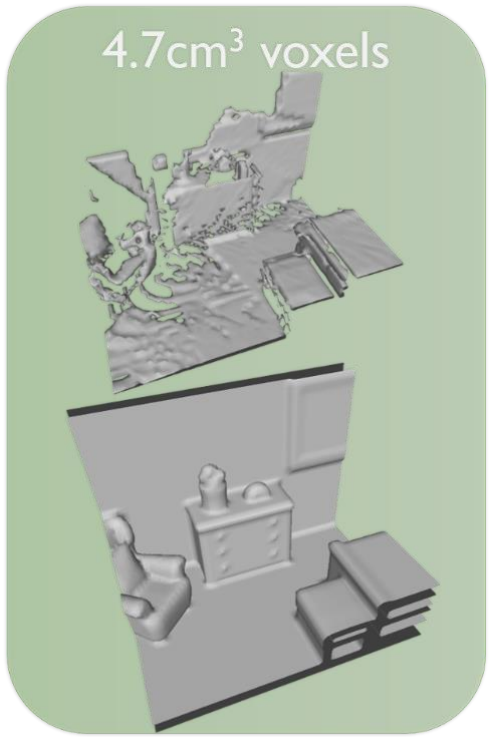


github.com/vdumoulin/conv_arithmetic

ScanComplete: Fully-convolutional

- Fully-convolutional network architecture for geometric completion

Train on crops of scenes

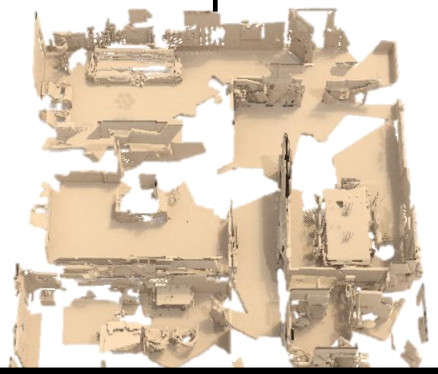


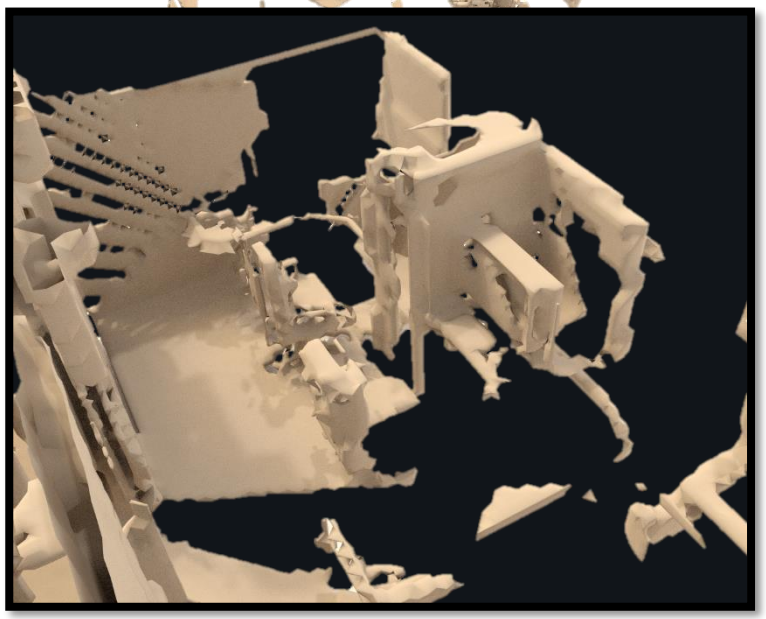
Test on entire scenes

[Dai et al. '18]

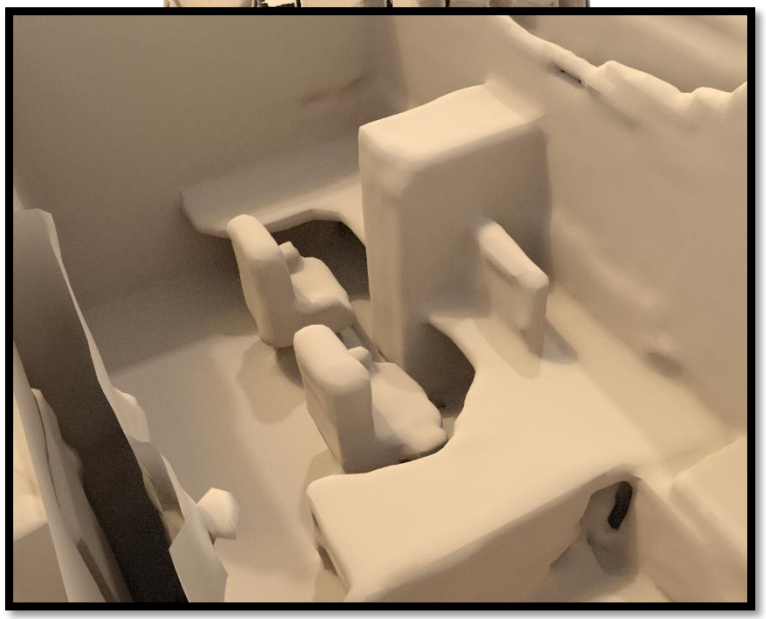
ScanComplete

Input

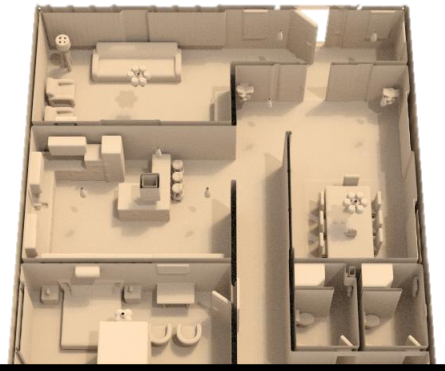


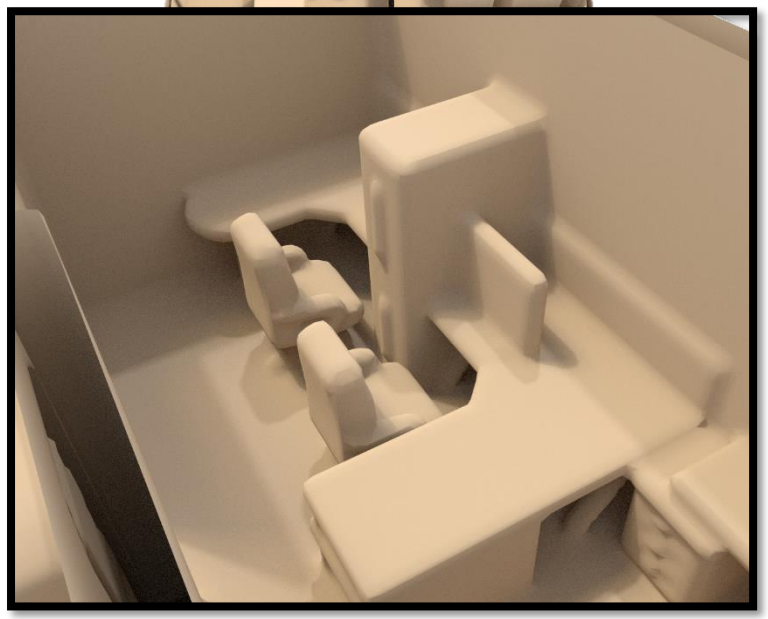


Completion



Ground Truth



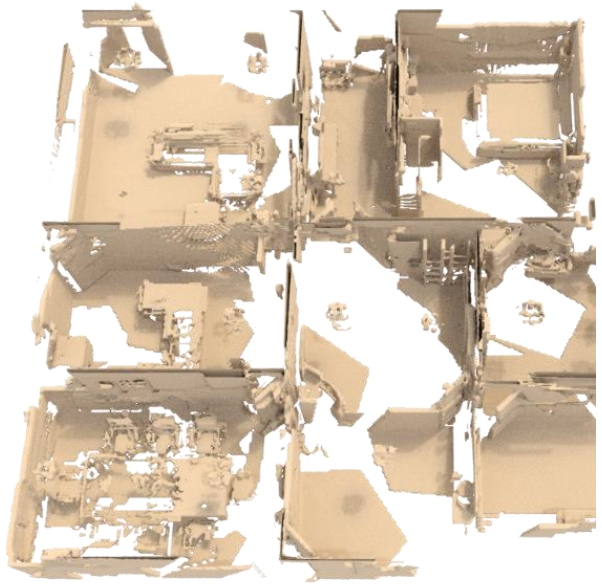


[Dai et al. '18]

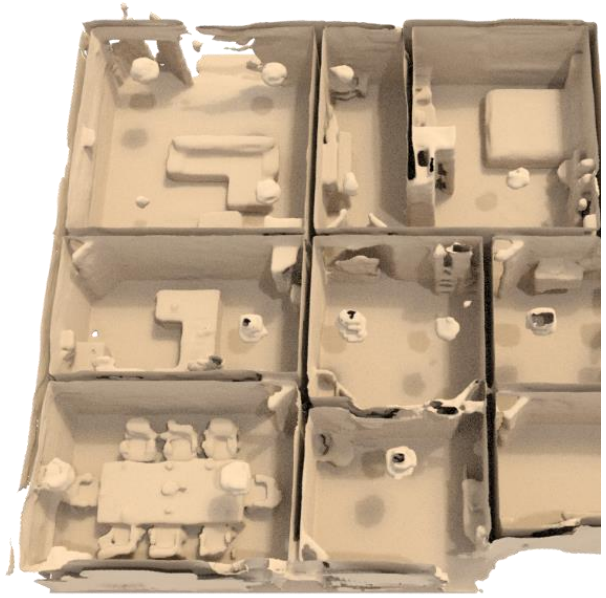
ScanComplete

- Predicting complete geometry helps with improved semantic segmentation

Input



Completion



Semantics

[Dai et al. '18]

ScanComplete

- Predicting complete geometry helps with improved semantic segmentation

Method	avg class accuracy
ScanNet [Dai et al. 17]	46.5
SSCNet [Song et al. 17]	59.6
ScanComplete (semantic-only)	56.2
ScanComplete (completion+semantics)	**71.3**

[Dai et al. '18]

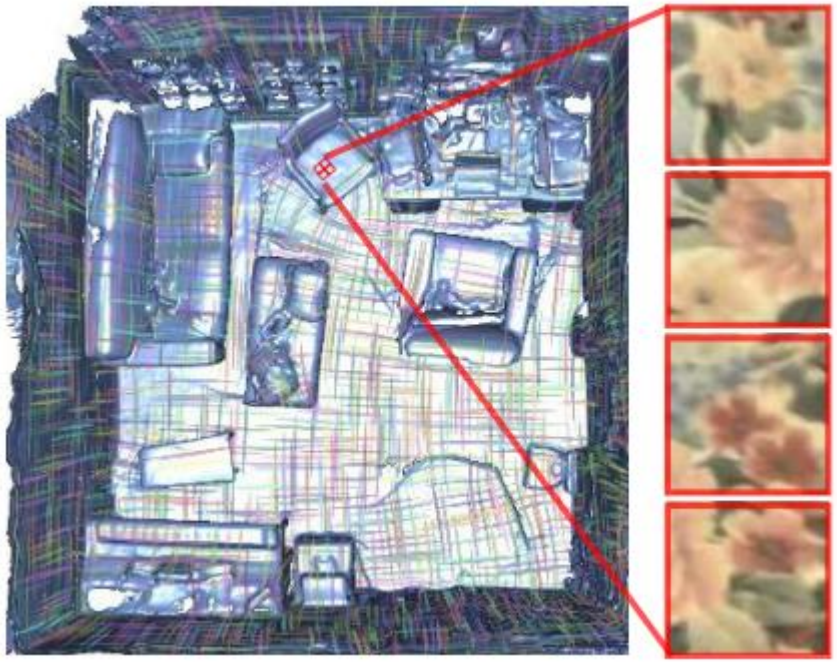
3D Semantic Segmentation

- Previously: operating at ≈5cm resolutions
- Often: memory limit
- Higher resolution: easier to distinguish many objects
- Imagine: 5cm pixel size
- How to operate at higher resolutions?

TextureNet

- Consider higher-resolution texture data vs geometry

Textured Mesh

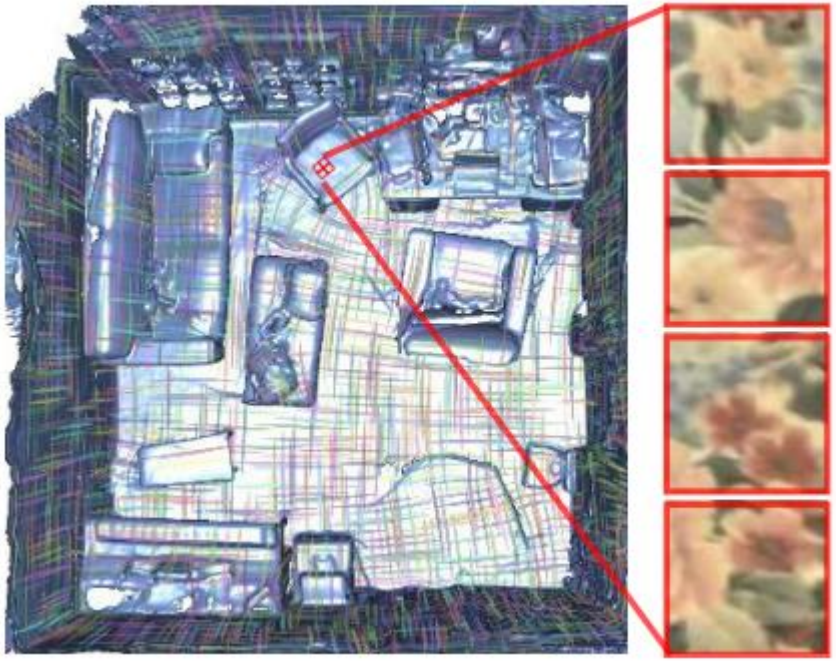


Higher-Resolution Texture Patches at each point

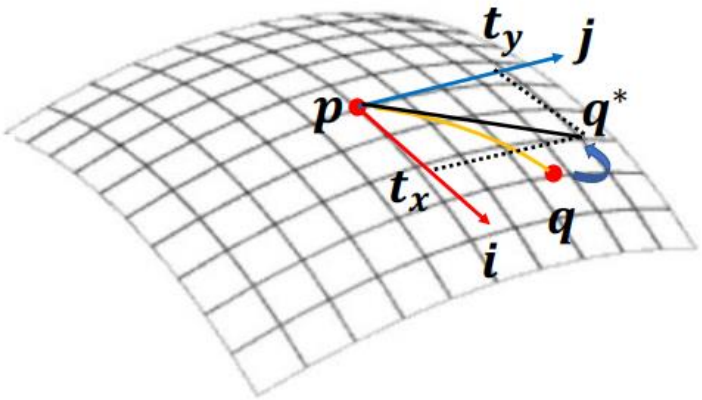
[Huang et al. ‘19]

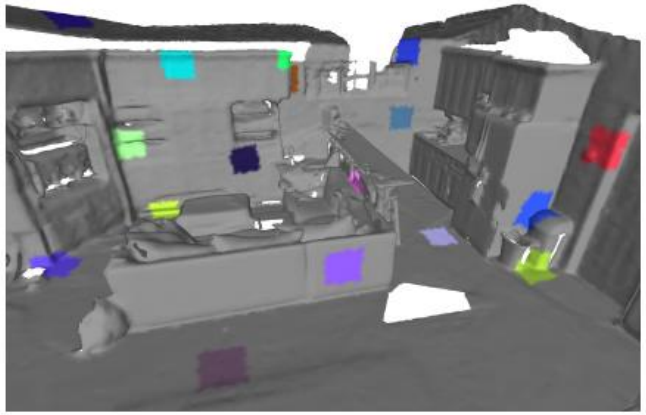
TextureNet

- Consider higher-resolution texture data vs geometry



Higher-Resolution Texture Patches at each point

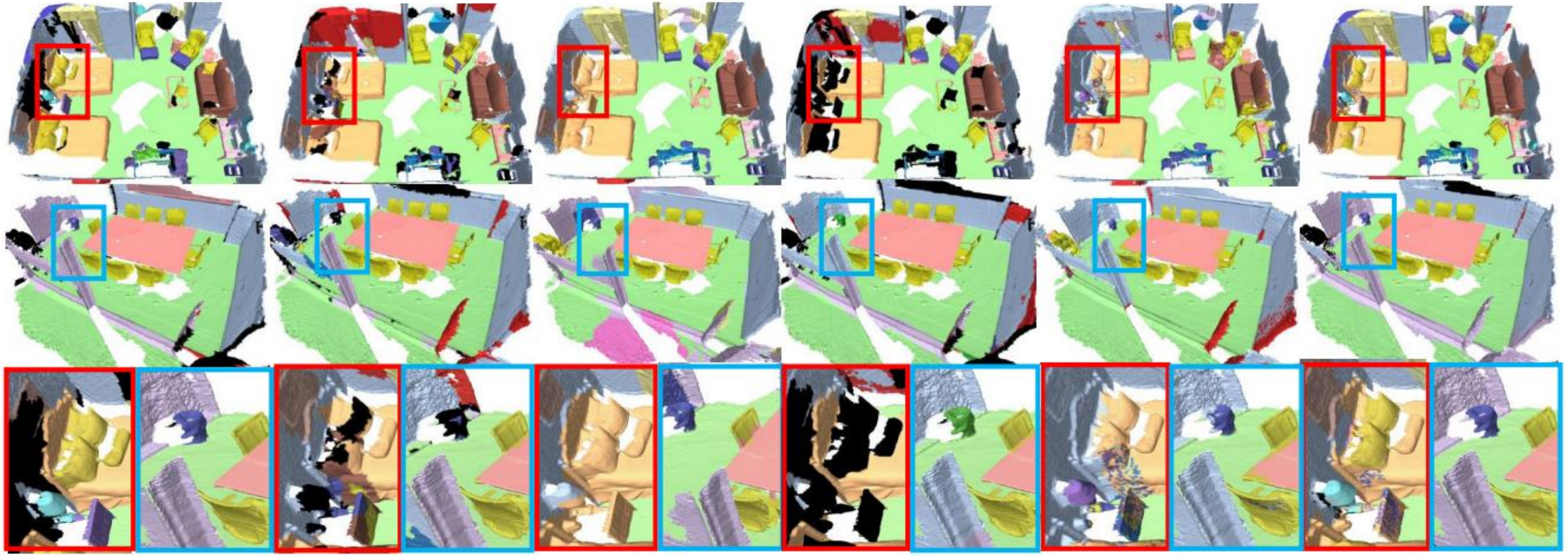




Convolutions over local geodesic neighborhoods.
Sample texture patches by geodesic neighborhood for convolution input.

[Huang et al. '19]

TextureNet



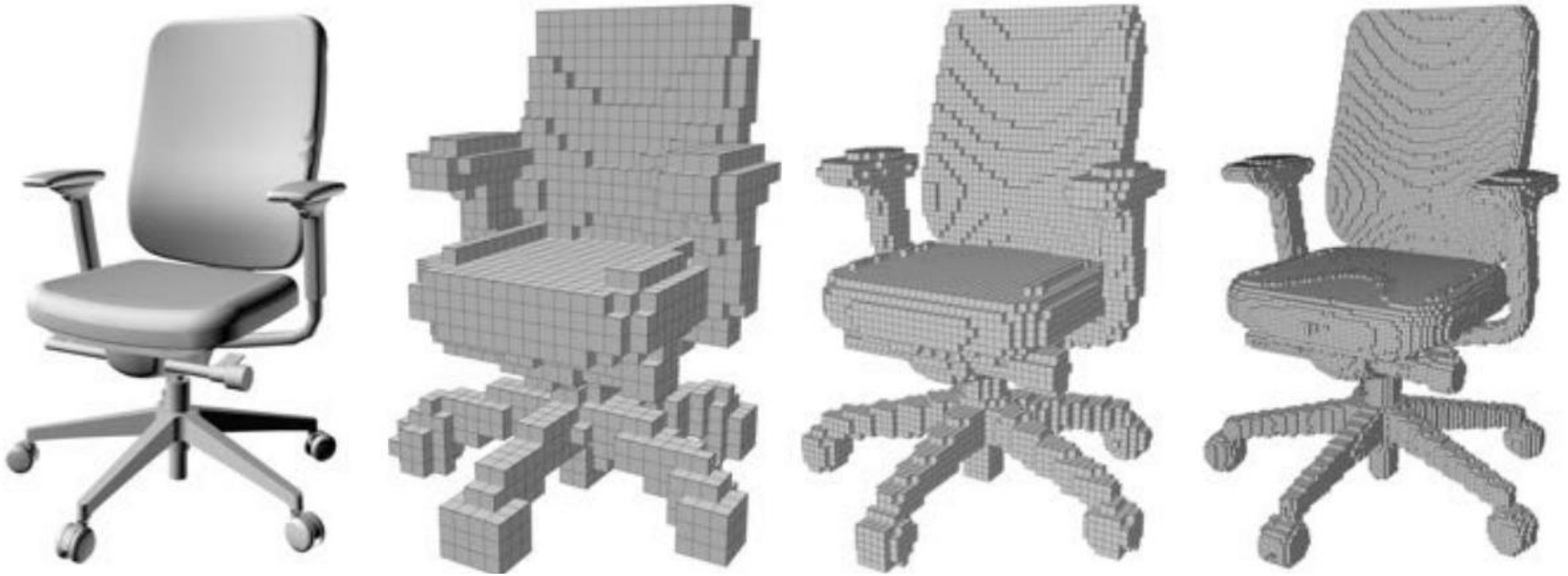
[Huang et al. '19]

Sparse Convolutions

- Can we make convolutions on a regular grid more efficient?

Sparse Convolutions

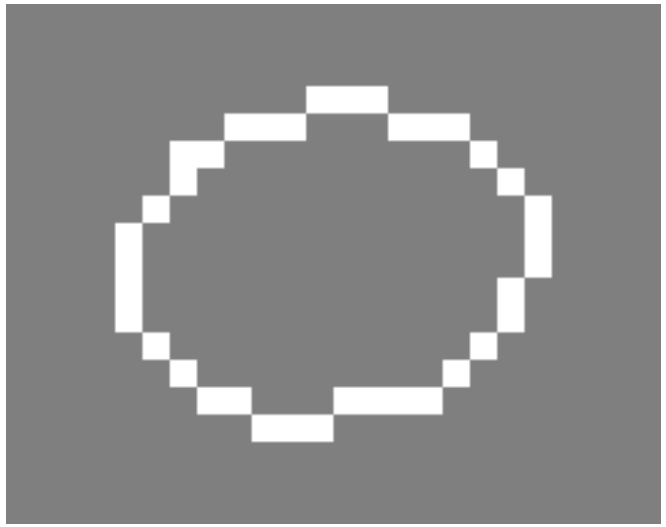
- Recall: Efficiency of dense volumetric representations



Percentage Occupancy:	10.4%	5.1%	2.4%
	32^3	64^3	128^3

Sparse Convolutions

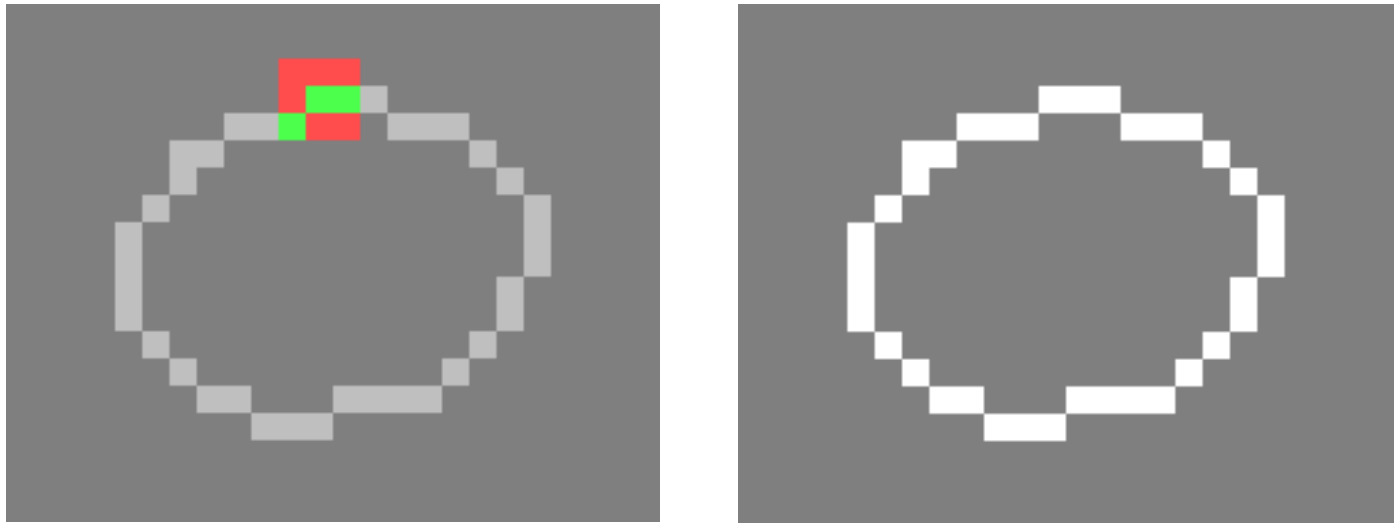
- Convolutions on a regular grid:
- Even with very sparse input, dense convolutions rapidly grow the active sites as the network grows deeper



[Graham et al. ‘17; Graham et al. ‘18]

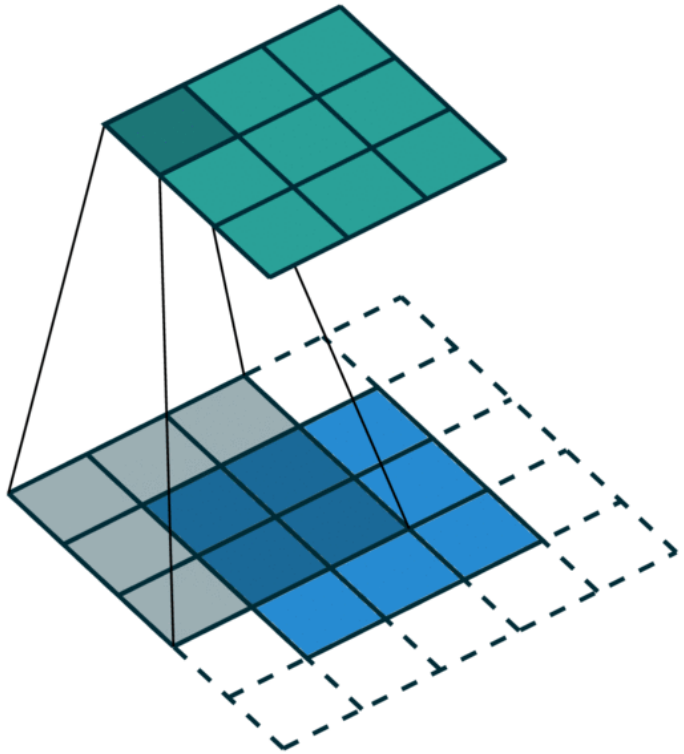
Sparse Convolutions

- Convolutions on active sites only

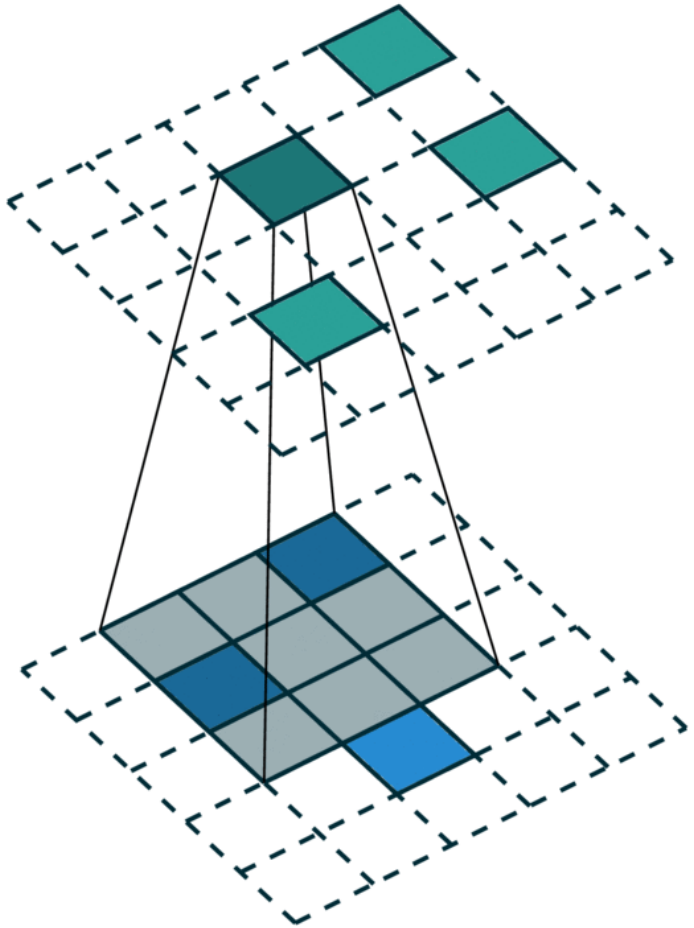


[Graham et al. ‘17; Graham et al. ‘18]

Sparse Convolutions



Regular Dense Convolution

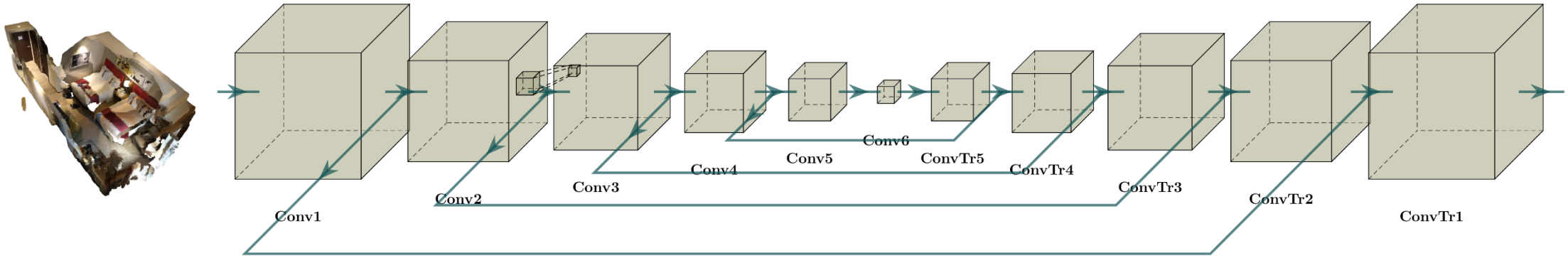


Sparse Convolution

[Graham et al. ‘17; Graham et al. ‘18; Choy et al. ‘19]

Sparse Convolutions

- Design analogous sparse convolutional architectures



- Operate on point clouds by discretizing into grid structure

[Graham et al. '17; Graham et al. '18; Choy et al. '19]

Sparse Convolutions

- Enables processing full scenes at much higher resolution (2cm, 1cm)

Method	Info	avg iou	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	de
MinkowskiNet	P	**0.734** 1	0.858 2	**0.833** 1	0.834 2	0.716 2	0.855 2	0.459 3	**0.836** 1	0.63
C. Choy, J. Gwak, S. Savarese: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. CVPR 2019										
SparseConvNet		0.725 2	0.647 14	0.821 2	**0.846** 1	**0.721** 1	**0.869** 1	**0.533** 1	0.754 4	0.60
joint point-based	P	0.634 5	0.614 16	0.778 3	0.667 11	0.633 4	0.825 3	0.420 7	0.804 2	0.467
Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, Winston H. Hsu: A Unified Point-Based Framework for 3D Segmentation. 3DV 2019										
MCCNN	P	0.633 6	**0.866** 1	0.731 5	0.771 4	0.576 8	0.809 5	0.410 9	0.684 8	0.49
P. Hermosilla, T. Ritschel, P.P. Vazquez, A. Vinacua, T. Ropinski: Monte Carlo Convolution for Learning on Non-Uniformly Sampled Point Clouds. SIGGRAPH A										
HPEIN		0.618 7	0.729 8	0.668 12	0.647 12	0.597 5	0.766 9	0.414 8	0.680 9	0.52
DMC-Net		0.608 8	0.732 7	0.729 6	0.694 6	0.536 9	0.783 6	0.427 6	0.639 13	0.438
LAP-D		0.594 9	0.720 9	0.692 10	0.637 14	0.456 16	0.773 8	0.391 12	0.730 5	0.58
DPC		0.592 10	0.720 9	0.700 8	0.602 17	0.480 12	0.762 10	0.380 14	0.713 6	0.58
Francis Engelmann, Theodora Kontogianni, Bastian Leibe: Dilated Point Convolutions. arXiv										
CCRFNet		0.589 11	0.766 5	0.659 15	0.683 8	0.470 15	0.740 12	0.387 13	0.620 14	0.490
TextureNet	P	0.566 12	0.672 12	0.664 13	0.671 9	0.494 10	0.719 14	0.445 4	0.678 11	0.411

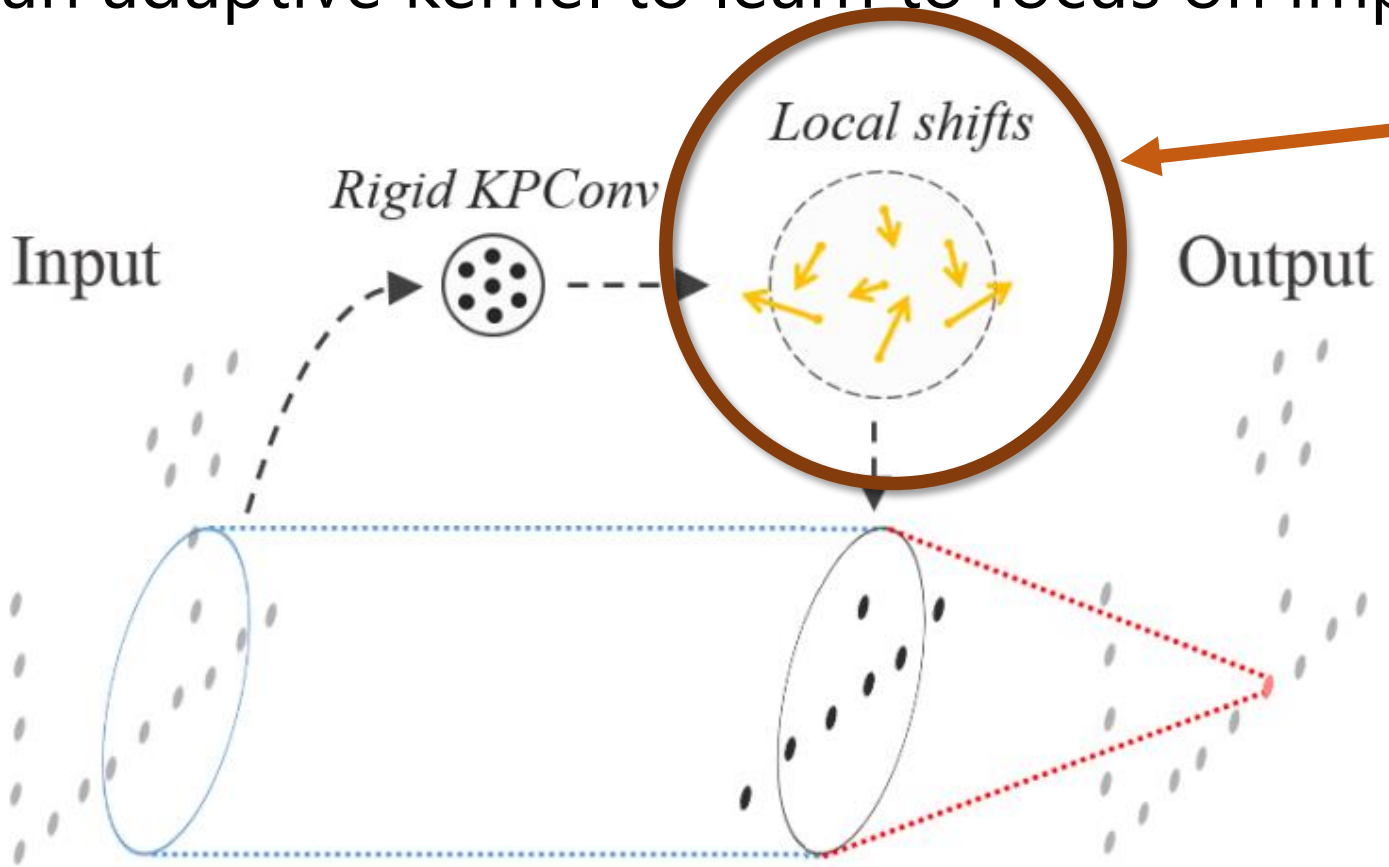
Sparse Convolutions

ScanNet Benchmark, June 2019

[Graham et al. '17; Graham et al. '18; Choy et al. '19]

KPConv

- Create an adaptive kernel to learn to focus on important regions only



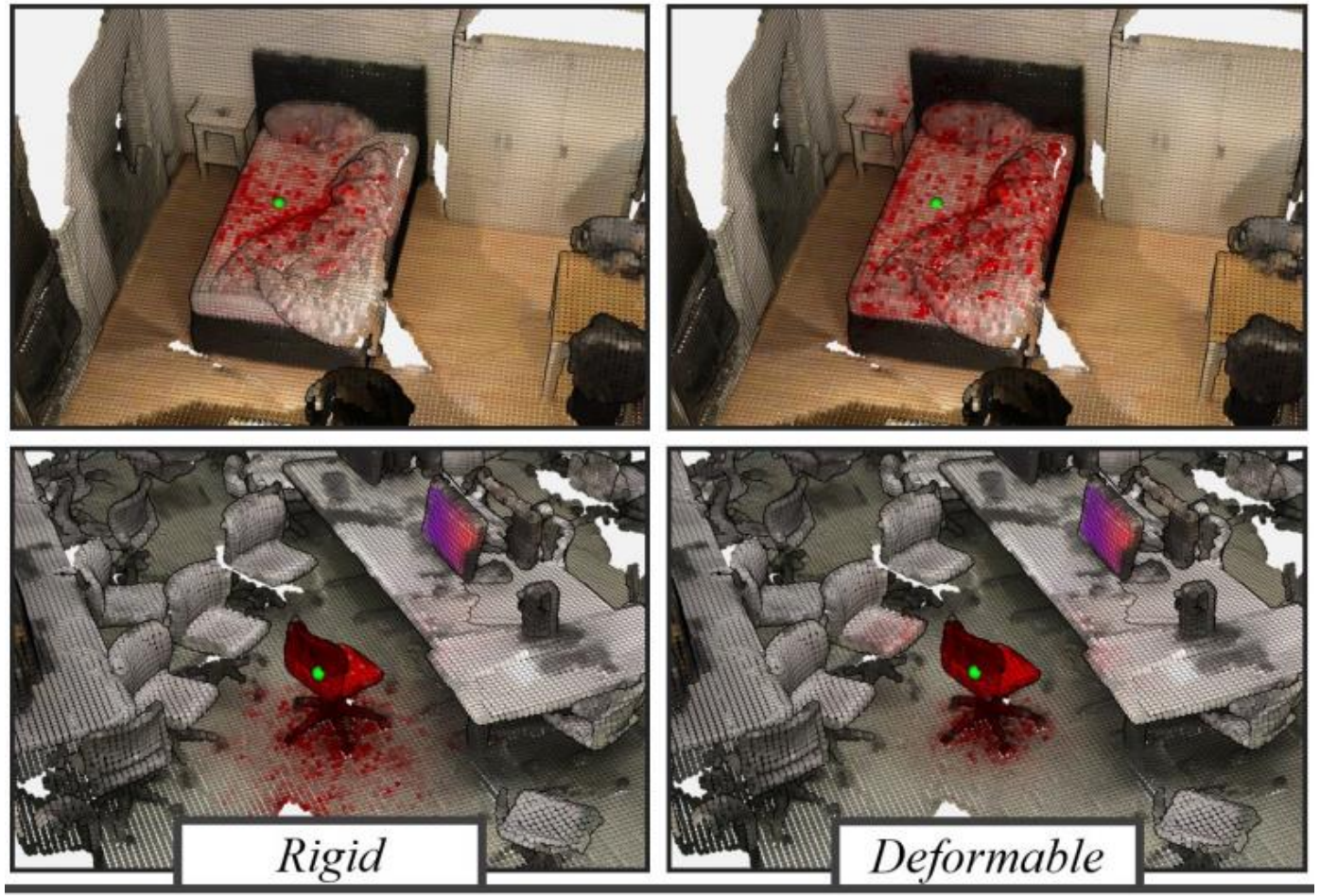
Predict offsets where kernel should be applied

Regularize offsets to stay near points to avoid "lost" kernels

[Thomas et al. '19]

KPConv

- Create an adaptive kernel to learn to focus on important regions only

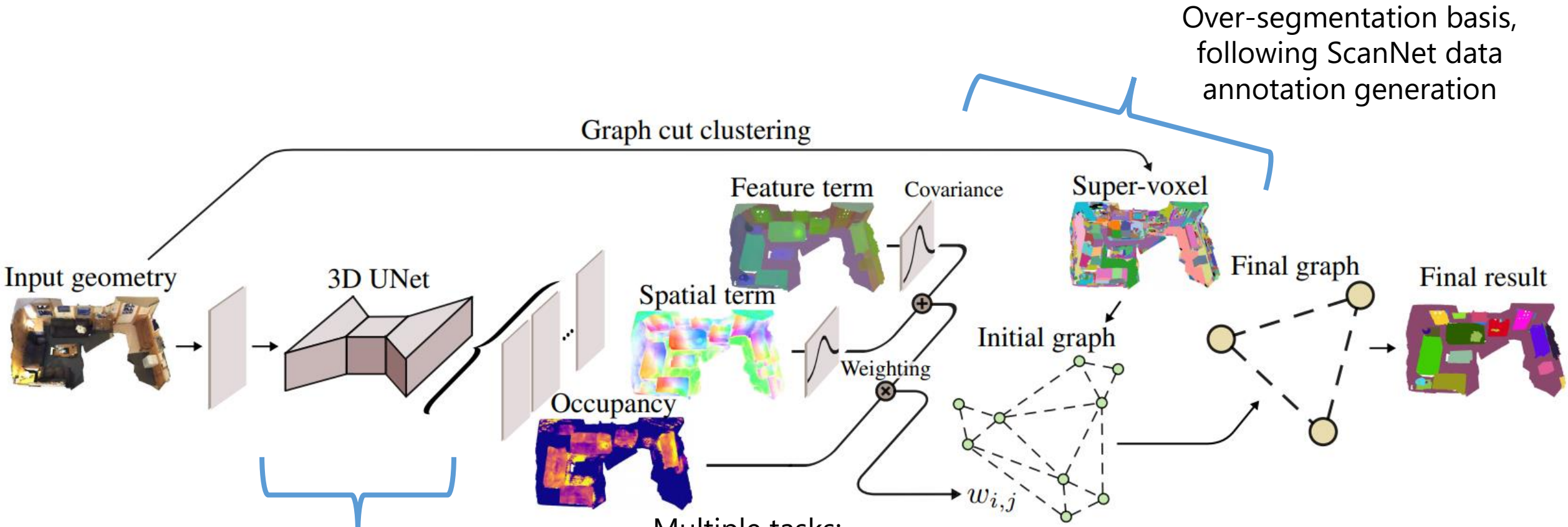


[Thomas et al. ‘19]

OccuSeg: Online Semantic Segmentation

[Han et al. '20]

OccuSeg: Online Semantic Segmentation



[Han et al. '20]

Virtual MVFusion

- Fusing image views with 3D geometry
- Previously: 3DMV approach, fuse original color views
  - Limited set of views
  - Imperfect alignments between color images and 3D geometry (due to camera estimation, motion blur, etc)
  - Original color views often have view-dependent effects (e.g., lighting)

[Kundu et al. ‘20]

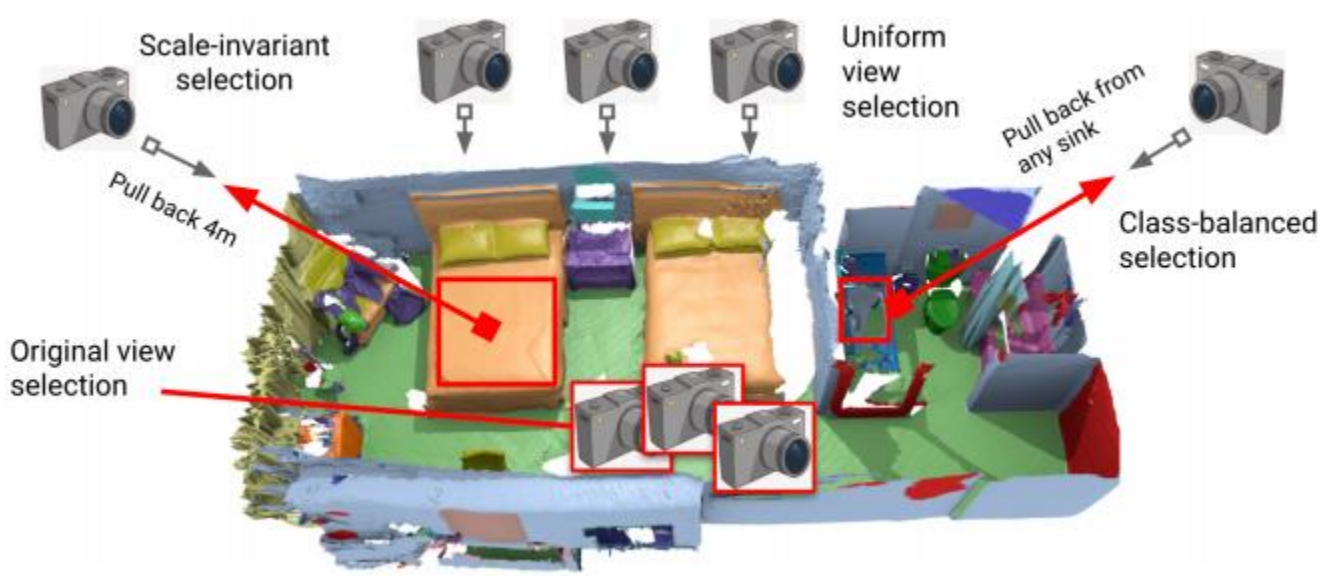
Virtual MVFusion

- Fusing image views with 3D geometry
- Use virtually selected viewpoints to render synthetic images of the 3D scene instead of using the original RGB images
  - Can create many more views
  - Can create images views with wider field of view
  - Views are consistent with each other (no view-dependent lighting effects, etc)
  - Views are consistent with 3D

[Kundu et al. ‘20]

Virtual MVFusion

- Fusing image views with 3D geometry
- Use virtually selected viewpoints to render synthetic images of the 3D scene instead of using the original RGB images



[Kundu et al. '20]

Virtual MVFusion

- Fusing image views with 3D geometry

- Use pre-trained 2D semantic segmentation
- Back-project all 2D features to 3D
- Aggregate all projected 2D features for a 3D point by average

- No explicit 3D convolutions

[Kundu et al. ‘20]

Virtual MVFusion

- Performance similar to 3D sparse convolution based

3D Semantic label benchmark

This table lists the benchmark results for the 3D semantic label scenario.

Method	Info	avg iou	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	floor	otherfurniture	picture	refrigerator
OccuSeg+Semantic		**0.764** 1	0.758 28	0.796 7	0.839 4	0.746 2	**0.907** 1	**0.562** 1	0.850 5	0.680 2	**0.672** 1	**0.978** 1	**0.610** 1	0.335 3	**0.777** 1
BPNet		0.749 2	**0.909** 1	0.818 4	0.811 8	**0.752** 1	0.839 6	0.485 12	0.842 7	0.673 3	0.644 4	0.957 3	0.528 7	0.305 9	0.773 2
Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, Tien-Tsin Wong: Bidirectional Projection Network for Cross Dimension Scene Understanding. CVPR 2021															
Virtual MVFusion		0.746 3	0.771 24	0.819 3	0.848 2	0.702 8	0.865 3	0.397 42	**0.899** 1	**0.699** 1	0.664 2	0.948 20	0.588 2	0.330 4	0.746 5
Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, Caroline Pantofaru: Virtual Multi-view Fusion for 3D Semantic Segmentation. ECCV 2020															
VMNet		0.746 3	0.870 5	**0.838** 1	**0.858** 1	0.729 4	0.850 4	0.501 7	0.874 2	0.587 19	0.658 3	0.956 4	0.564 4	0.299 10	0.765 3
MinkowskiNet	P	0.736 5	0.859 7	0.818 4	0.832 5	0.709 7	0.840 5	0.521 4	0.853 4	0.660 4	0.643 5	0.951 10	0.544 5	0.286 15	0.731 6
C. Choy, J. Gwak, S. Savarese: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. CVPR 2019															
SparseConvNet		0.725 6	0.647 46	0.821 2	0.846 3	0.721 5	0.869 2	0.533 2	0.754 18	0.603 16	0.614 6	0.955 5	0.572 3	0.325 5	0.710 7

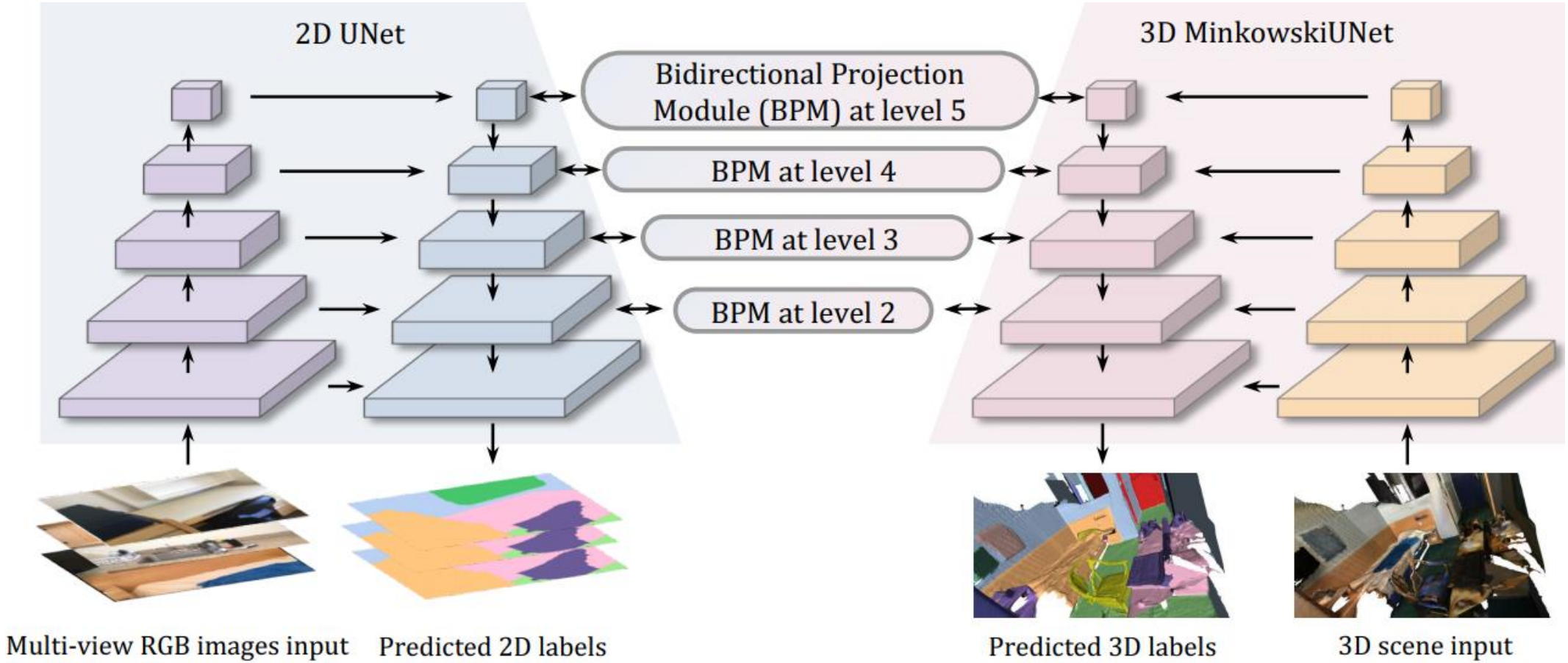
[Kundu et al. '20]

BPNet

- Joint 2D-3D fusion
- Previous: 2D features -> 3D
- BPNet: bidirectional interactions, 2D <-> 3D

[Hu et al. ‘21]

BPNet

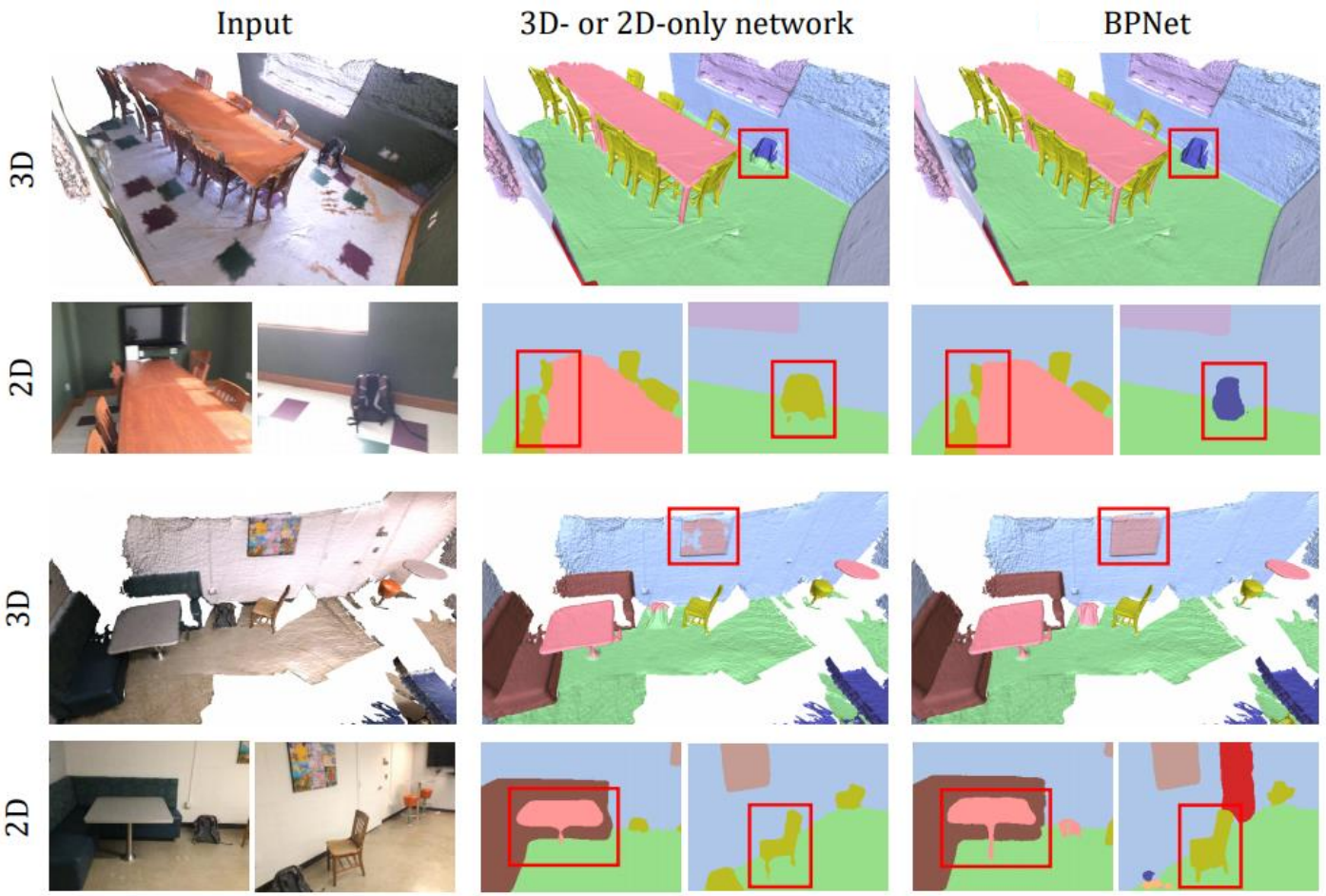


[Hu et al. '21]

BPNet

- Can improve 2D and 3D

Method	Projection Level	mIoU	
		2D	3D
UNet34	—	61.5	—
MinkowskiUNet18A	—	—	68.0
Ours W/ BPM	P2	63.5	70.3
Ours W/ BPM	P3	64.1	70.5
Ours W/ BPM	P4	62.3	69.7
Ours W/ BPM	P5	61.8	68.5
Ours W/ BPM	P2, P3, P4, P5	**65.1**	**70.6**
Ours W/ $UPM_{2D \rightarrow 3D}$	P2, P3, P4, P5	62.2	69.7
Ours W/ $UPM_{2D \leftarrow 3D}$	P2, P3, P4, P5	65.0	68.8



[Hu et al. '21]

Vision Transformers

- CNN inductive biases
  - Locality, weight sharing, translational equivariance

- Transformer: operates on sequences
  - No translation invariances
  - Permutation invariance
  - Key operator: attention

Attention

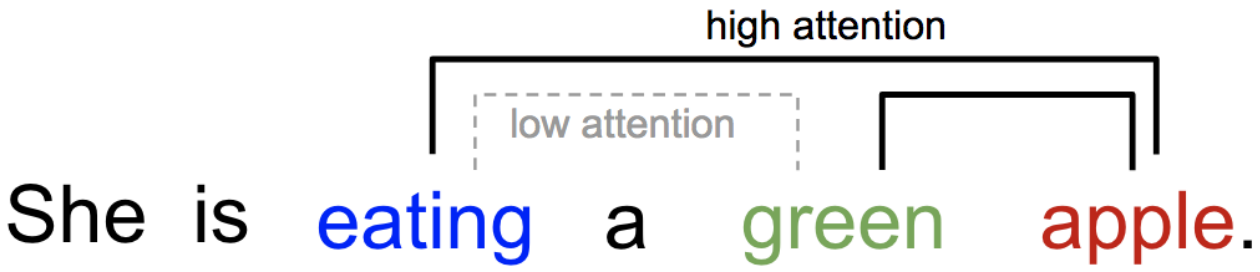
- Sequence-to-sequence task
- Token: element of sequence (e.g., word in sentence)

- Mimic cognitive attention
  - Focus more on some elements than others
  - Learn which elements are more important than others based on context

- Note: implicitly, deep networks can already learn some form of attention implicitly

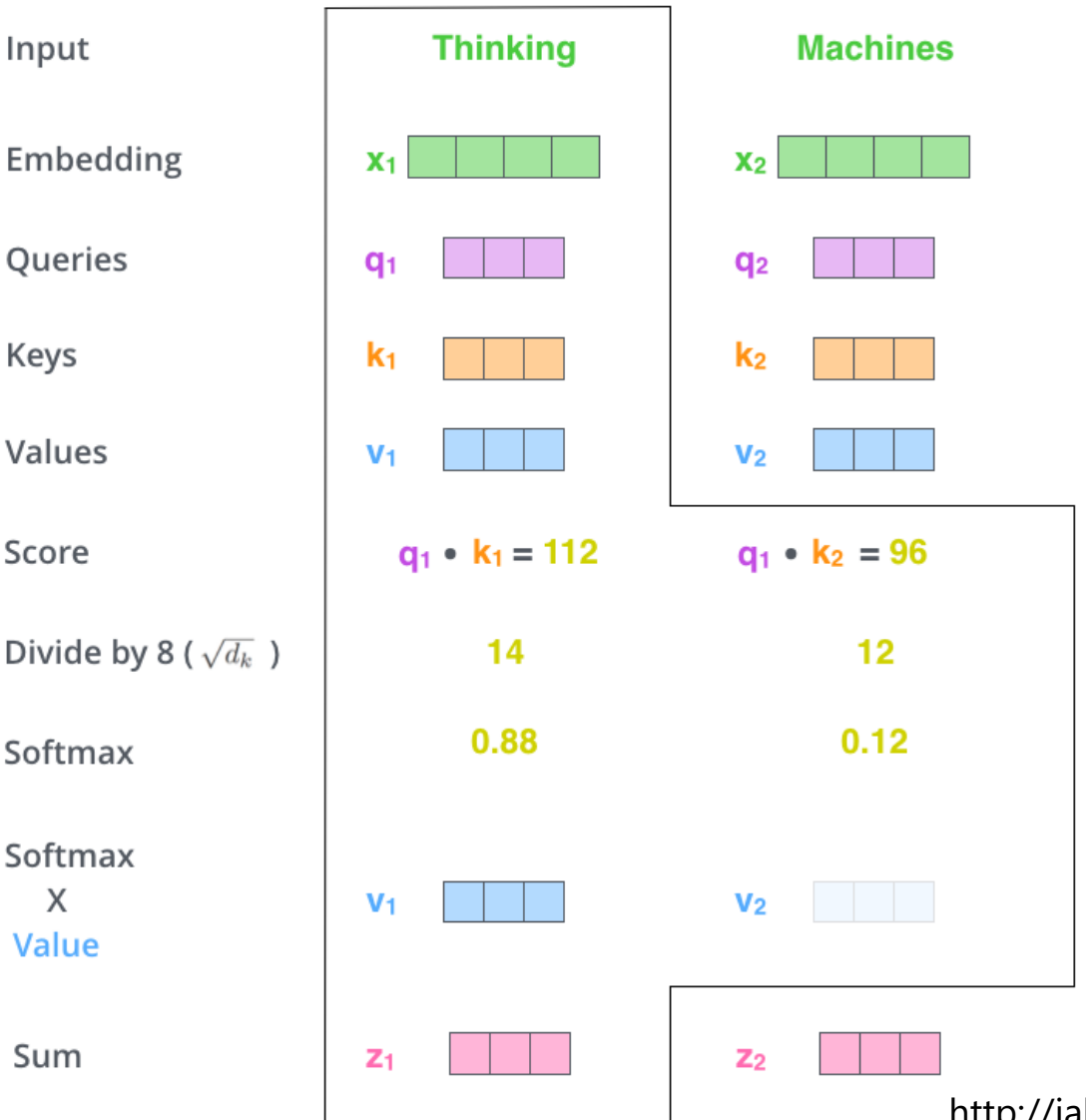
Attention

- Explicitly model sensitivity to input elements
- Key-value-query
  - Ex: youtube search
    - Query: text query
    - Keys: video features (e.g., title, description, etc)
    - Values: videos



https://lilianweng.github.io/posts/2018-06-24-attention/

Self-Attention



http://jalammar.github.io/illustrated-transformer/

Multi-headed Attention

1) This is our input sentence*

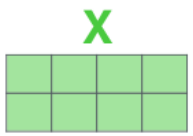
2) We embed each word*

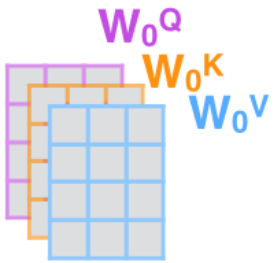
3) Split into 8 heads. We multiply X or R with weight matrices

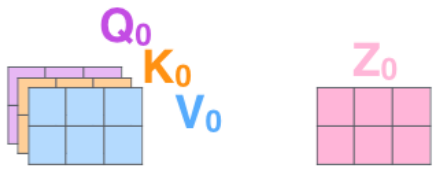
4) Calculate attention using the resulting Q/K/V matrices

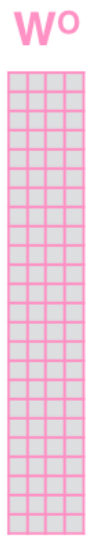
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

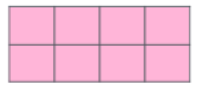
Thinking Machines



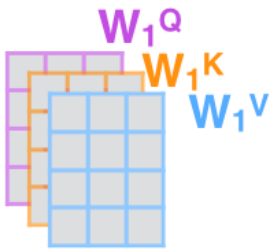


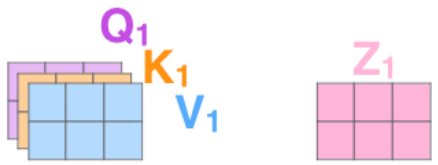


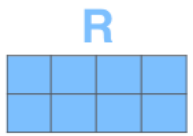




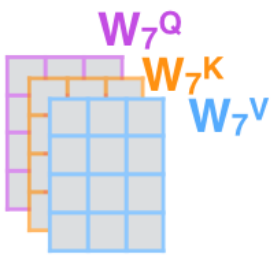
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

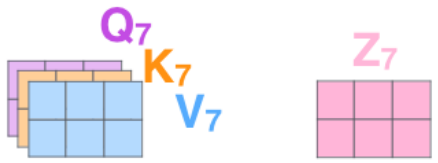






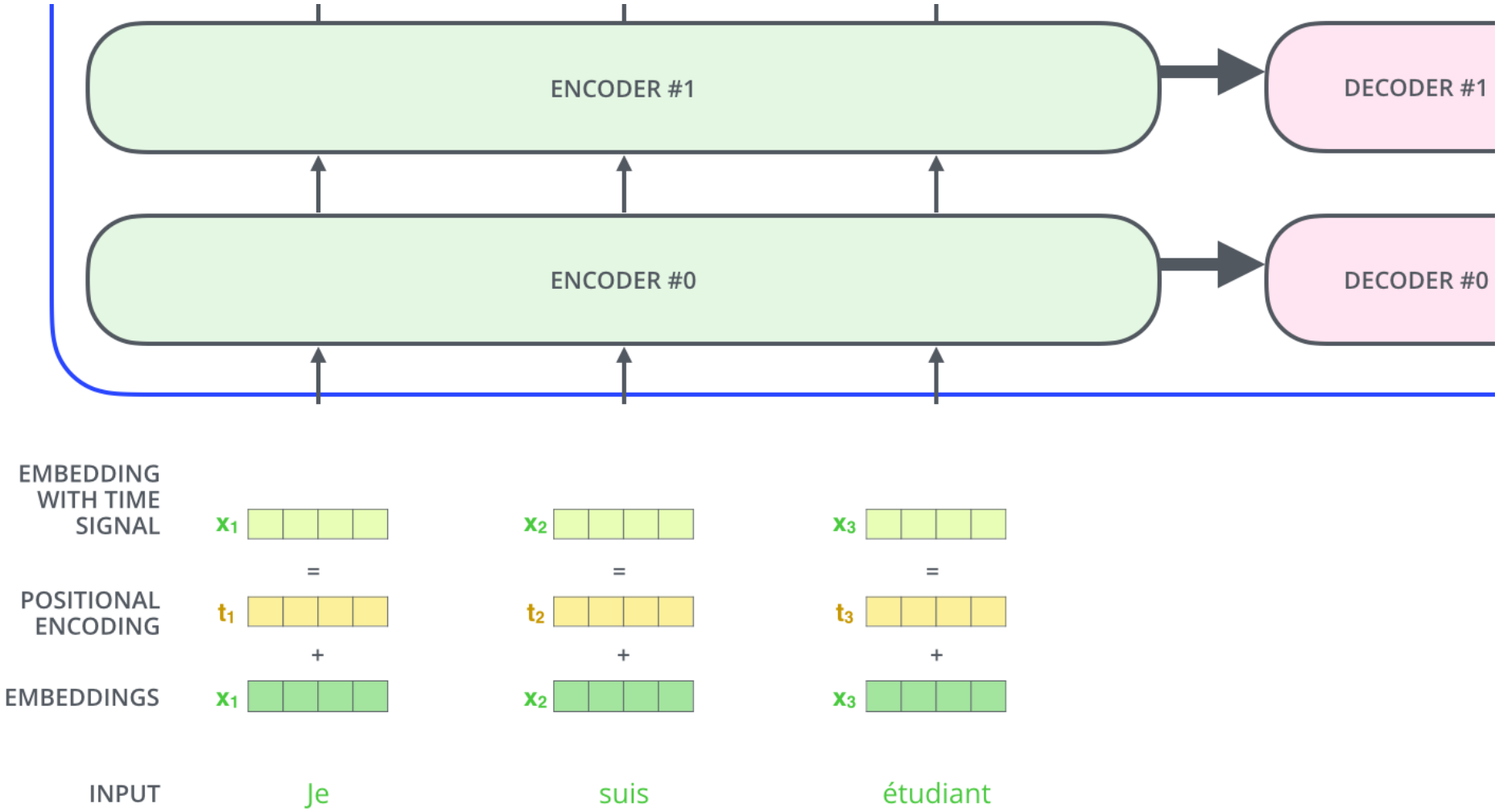
...





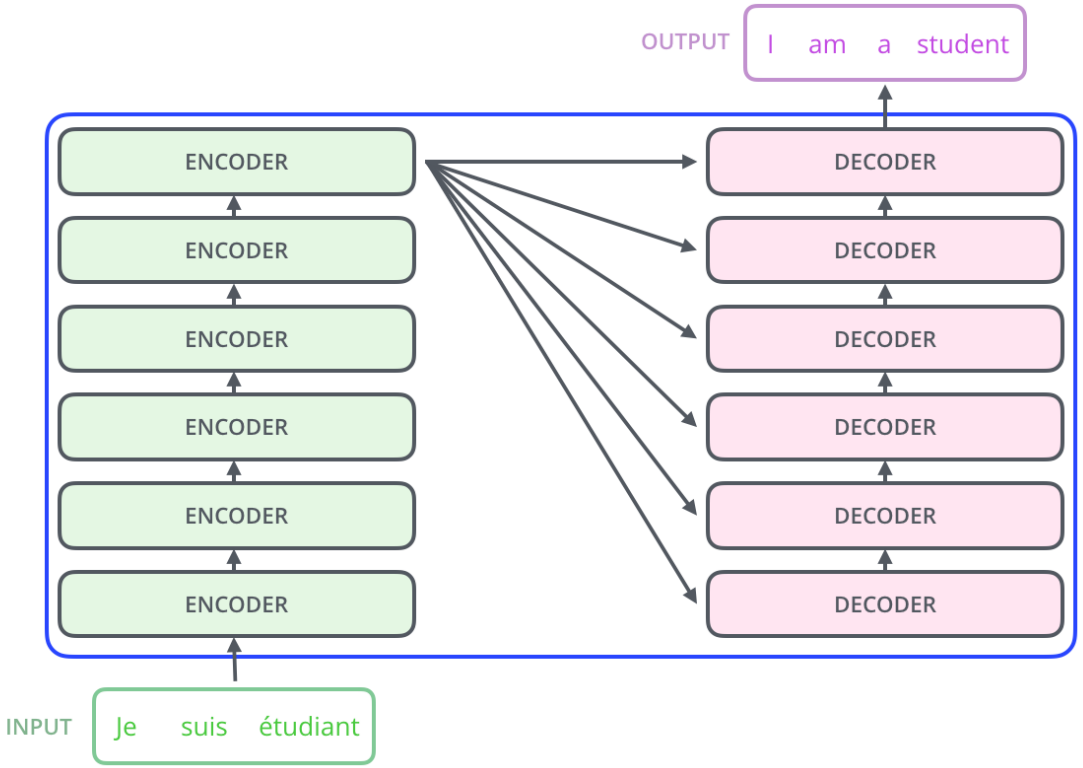
http://jalammar.github.io/illustrated-transformer/

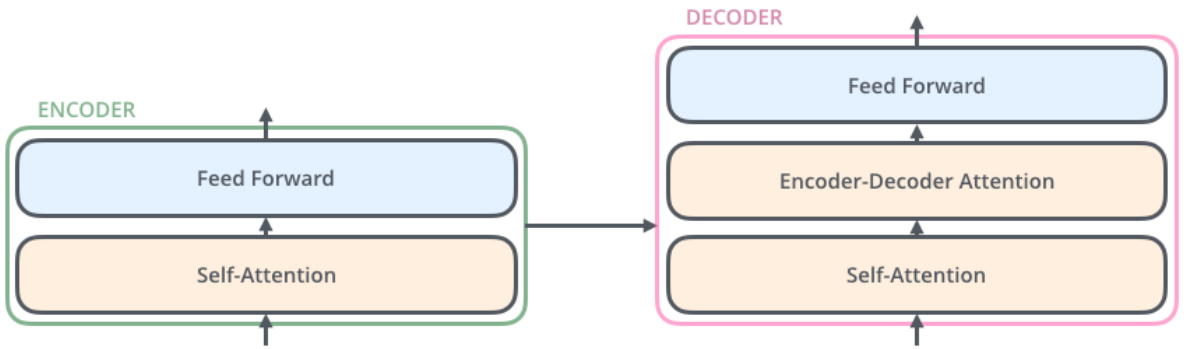
Positional Encoding



http://jalammar.github.io/illustrated-transformer/

Transformer



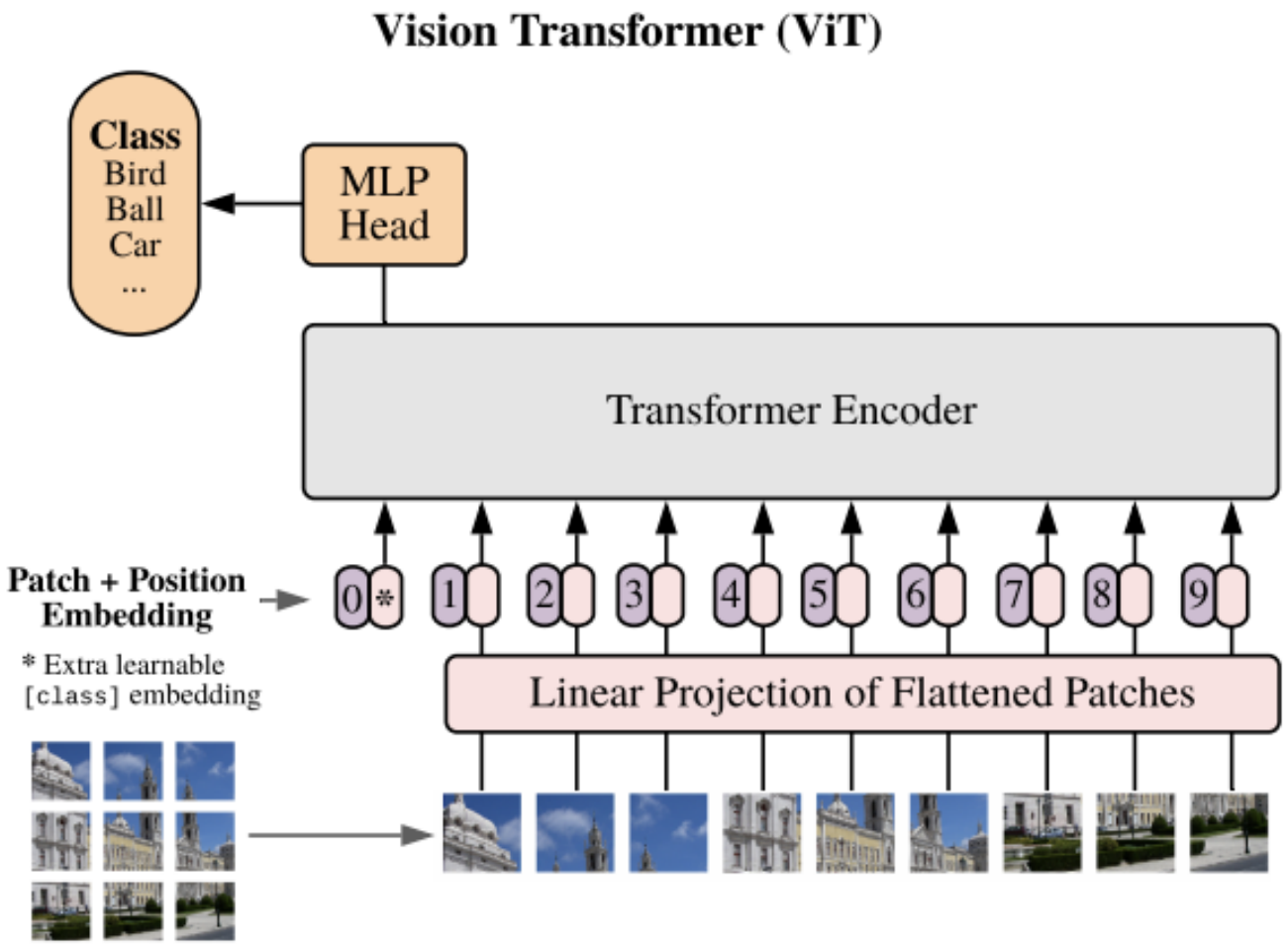


http://jalammar.github.io/illustrated-transformer/

Transformer

- Significant success in natural language processing
- The “new” convolutions
- Note: self-attention complexity wrt sequence length is high: $O(n^2 d)$
  - n: sequence length
  - d: representation dimension
- How to leverage for higher-dimensional data (2D, 3D)?

Vision Transformer



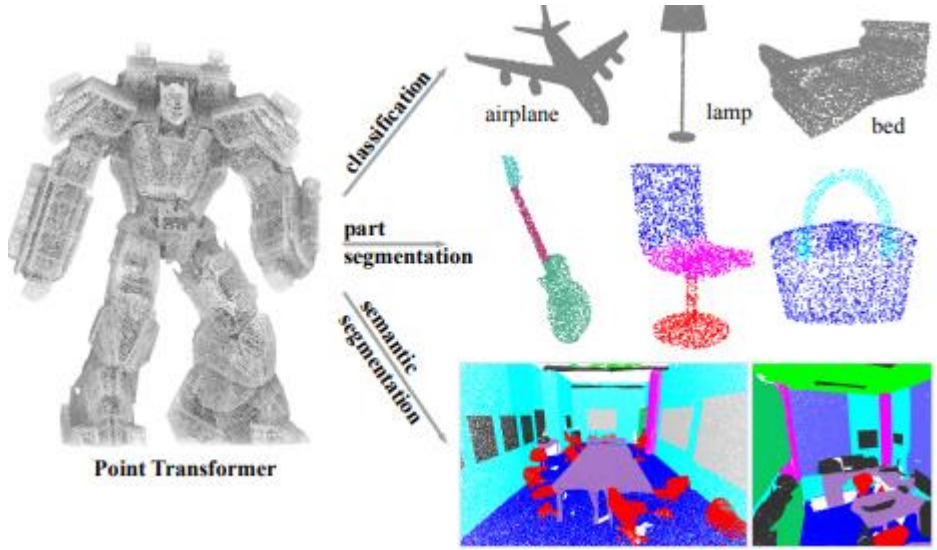
[Dosovitskiy et al. 21]

Transformers for 3D

Transformer on Point Clouds

- Self-attention is a set operator
  - Invariant to permutation and cardinality of inputs

- Dimensionality:
  - Use smaller number of points
  - Operate on clusters of points
    (e.g., similar to PointNet++ processing)



PointTransformer [Zhao et al. ‘21]

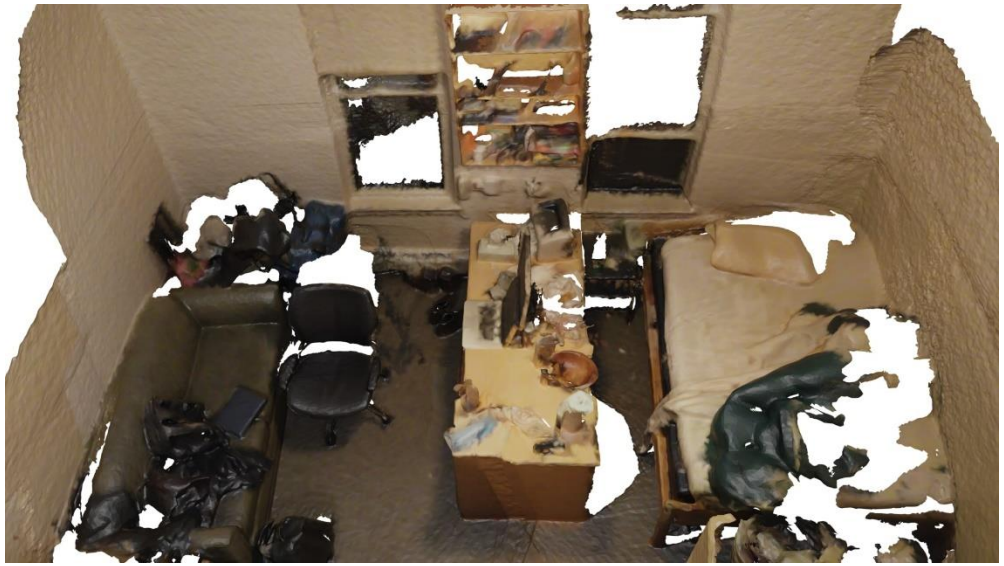
Transformer on Voxels

- Use sparse convolutions to create lower-resolution sparse set of features to operate on

- Pre-compute clusters of voxels to operate on
  - E.g., traditional clustering approach like Felzenszwalb

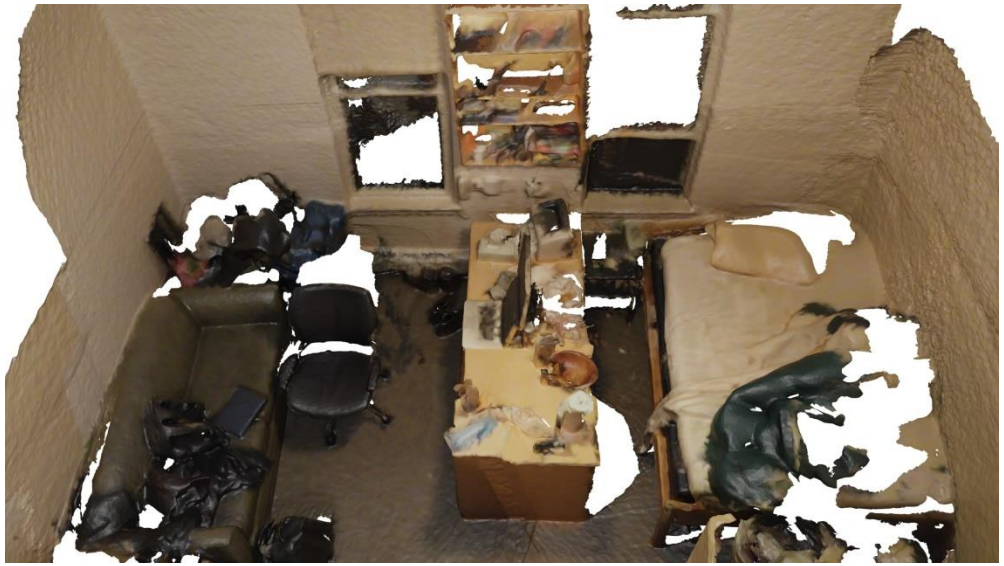
Large-Vocabulary 3D Understanding

- 3D scene understanding benchmarks: <30 classes



Large-Vocabulary 3D Understanding

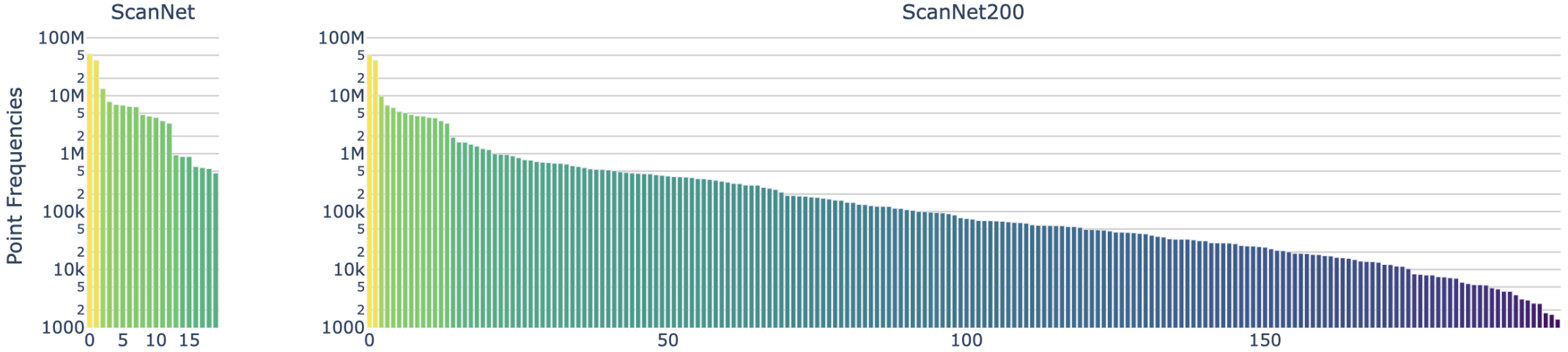
- Large-vocabulary 3D semantic understanding: 200 classes



ScanNet200 [Rozenberszki et al. ‘22]

Large-Vocabulary 3D Understanding

- Large-vocabulary 3D semantic understanding: 200 classes



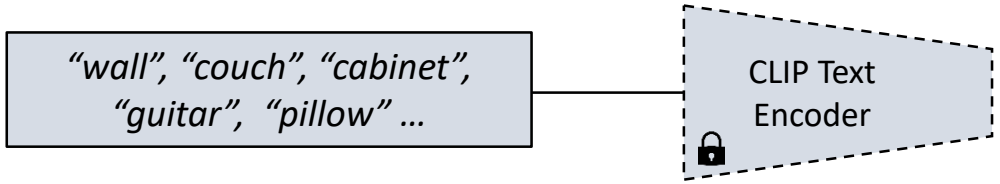
ScanNet200 [Rozenberszki et al. ‘22]

Language-Grounded 3D Learning

“wall”, “couch”, “cabinet”, “guitar”, “pillow” …

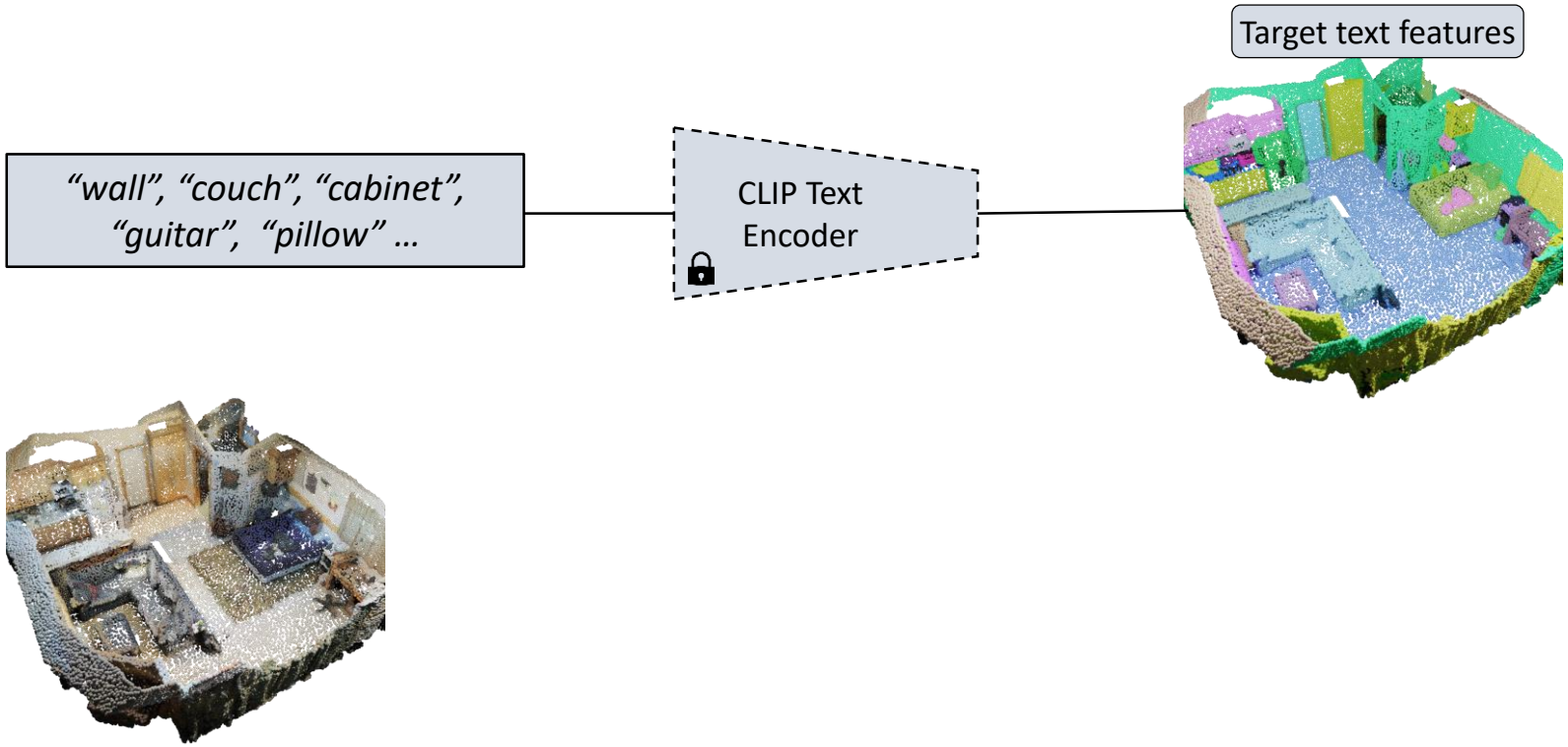
ScanNet200 [Rozenberszki et al. ‘22]

Language-Grounded 3D Learning



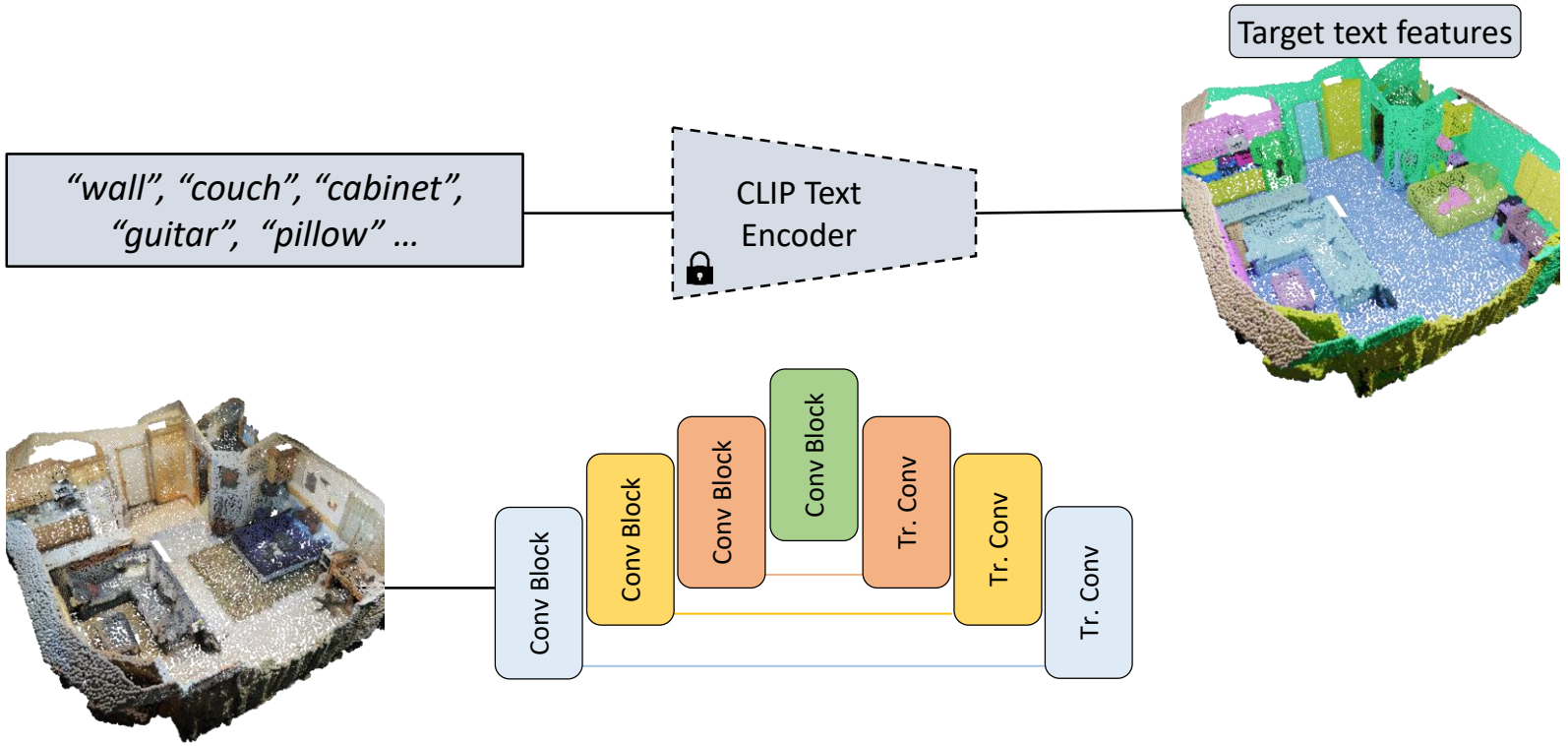
ScanNet200 [Rozenberszki et al. ‘22]

Language-Grounded 3D Learning



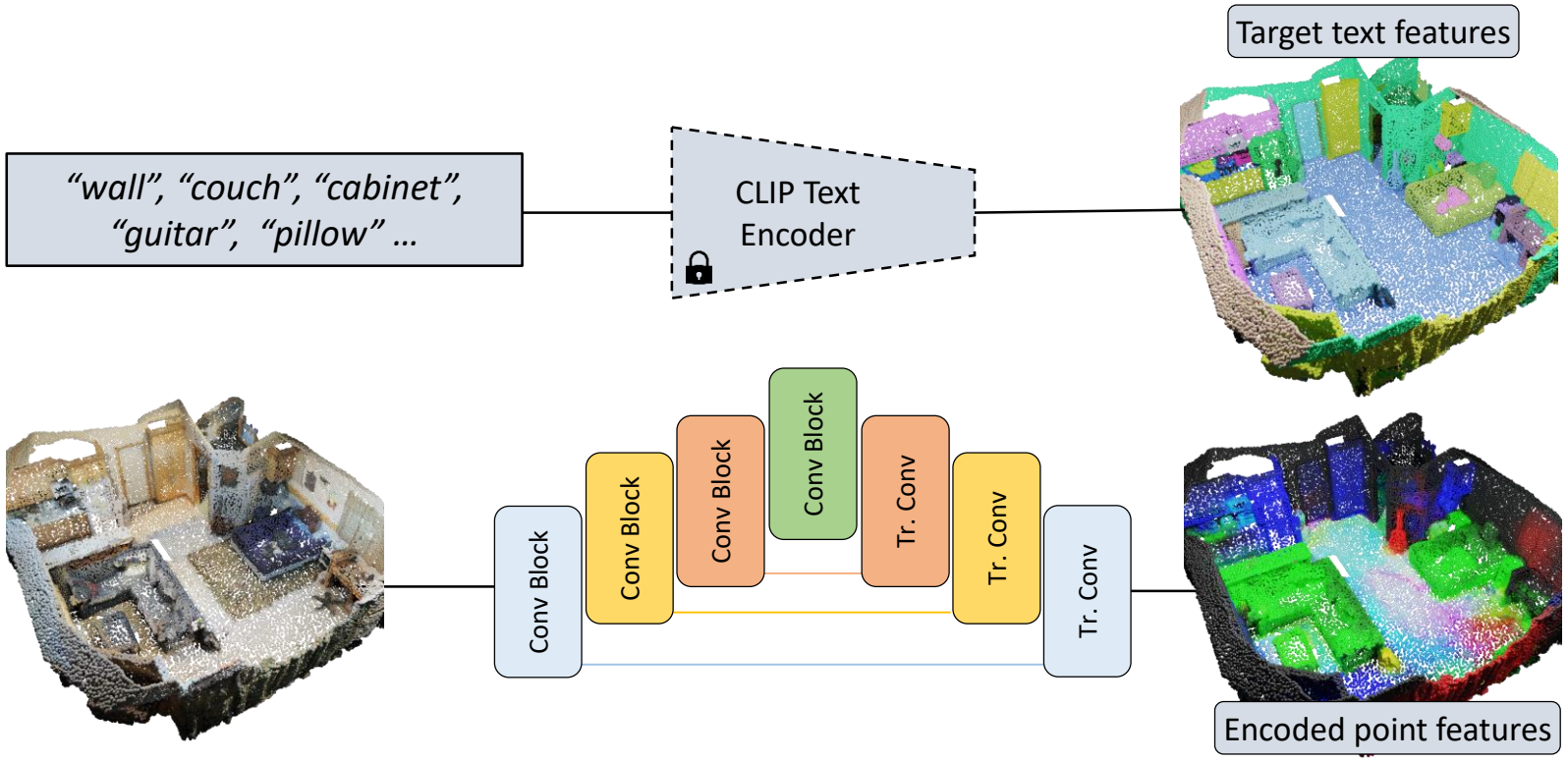
ScanNet200 [Rozenberszki et al. ‘22]

Language-Grounded 3D Learning



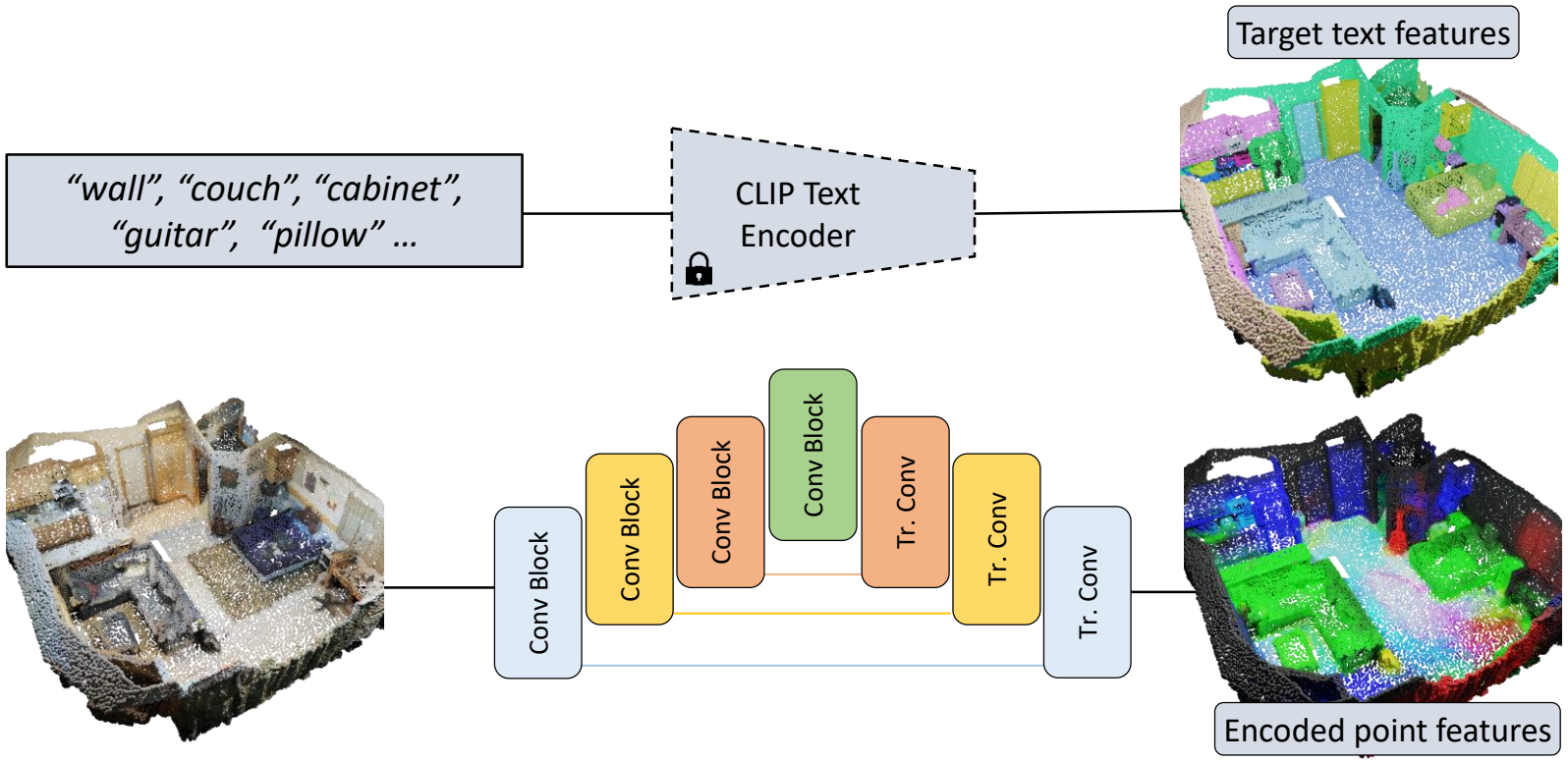
ScanNet200 [Rozenberszki et al. '22]

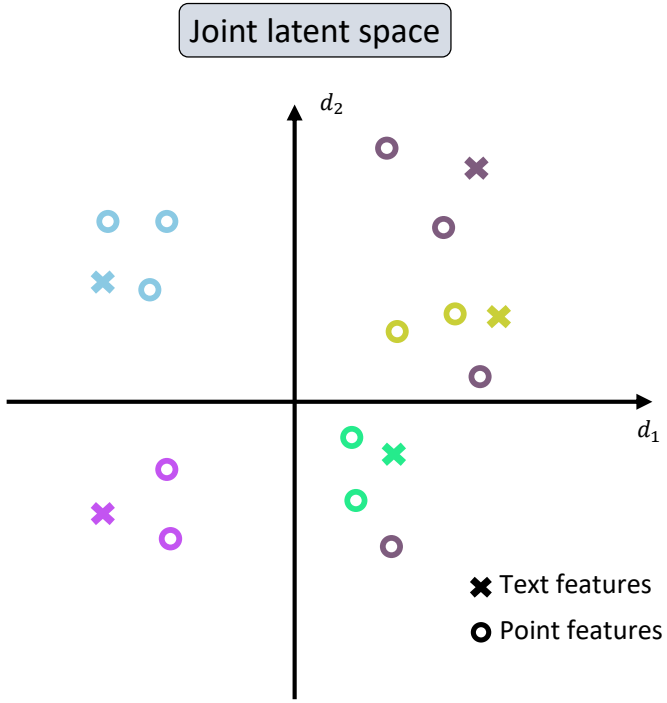
Language-Grounded 3D Learning



ScanNet200 [Rozenberszki et al. '22]

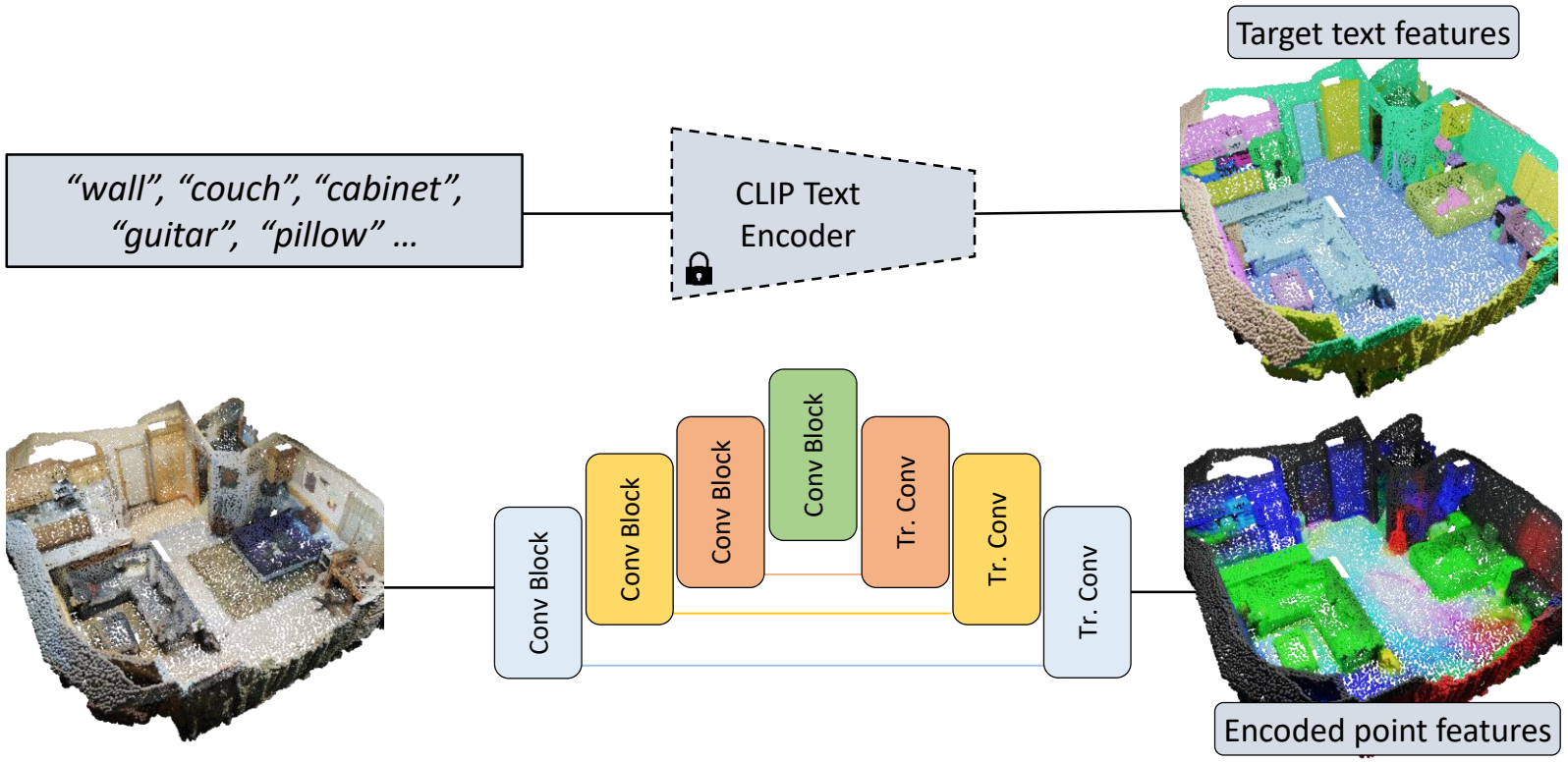
Language-Grounded 3D Learning

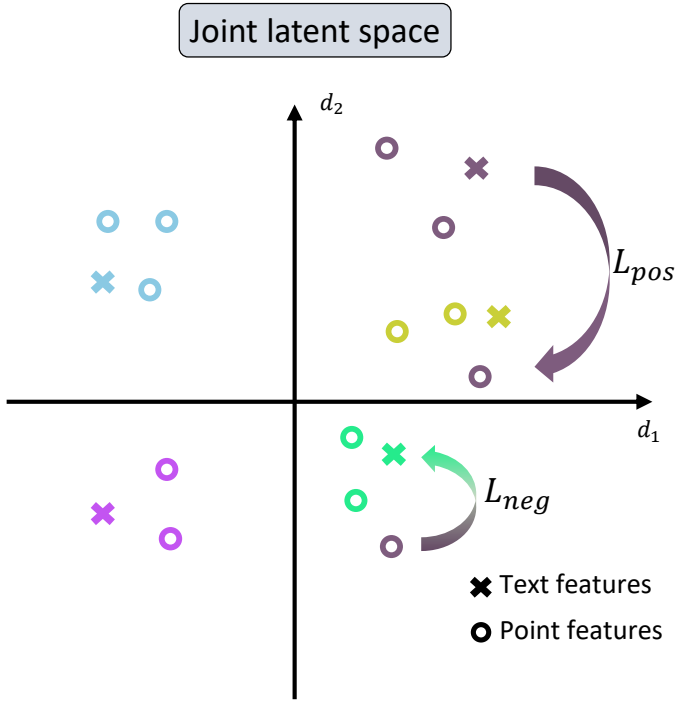




ScanNet200 [Rozenberszki et al. '22]

Language-Grounded 3D Learning

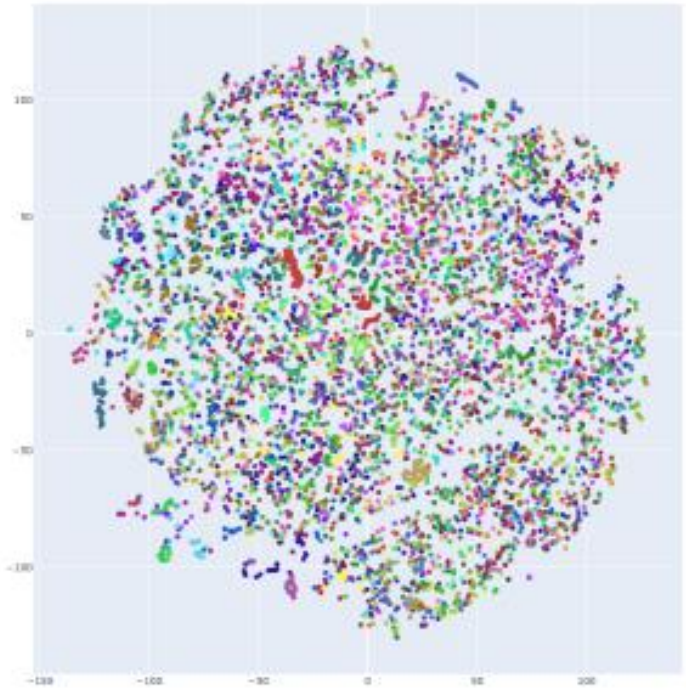




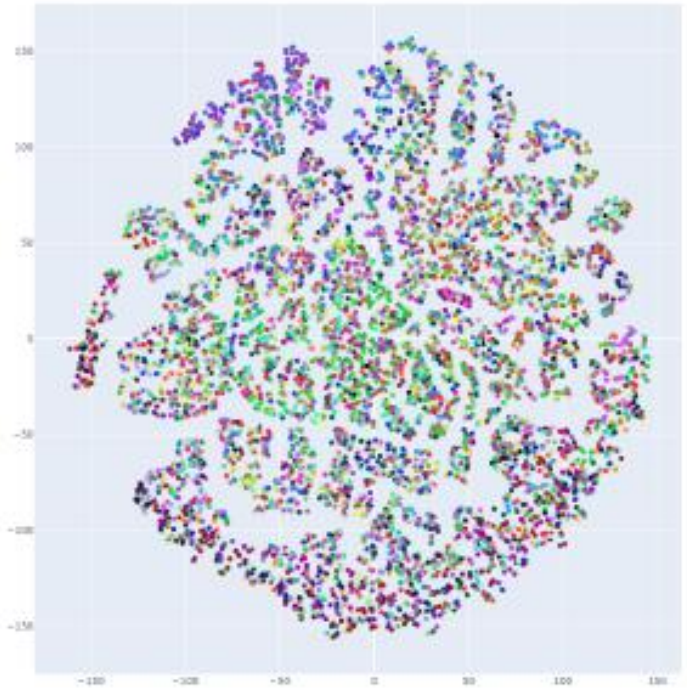
ScanNet200 [Rozenberszki et al. '22]

Language-Grounded 3D Learning

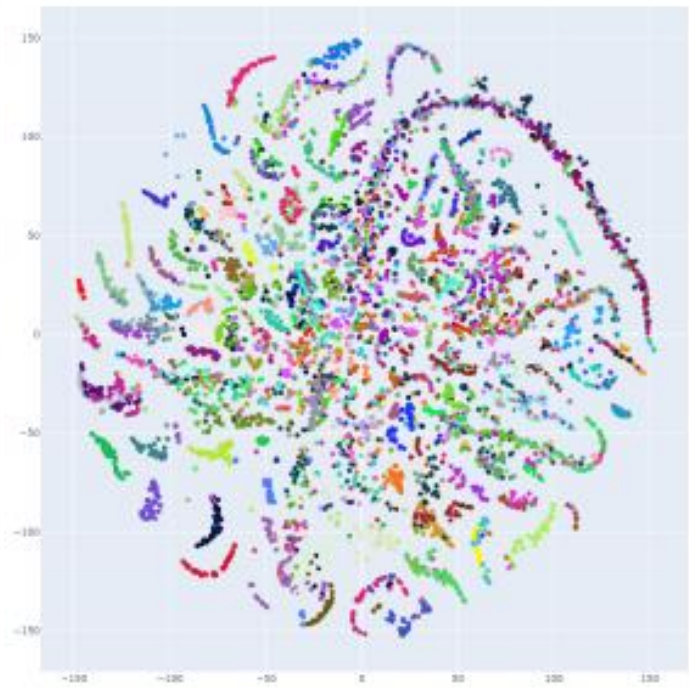
- Learned feature space



CSC [Hou et al. ‘21]



SupCon [Khosla et al. ‘20]

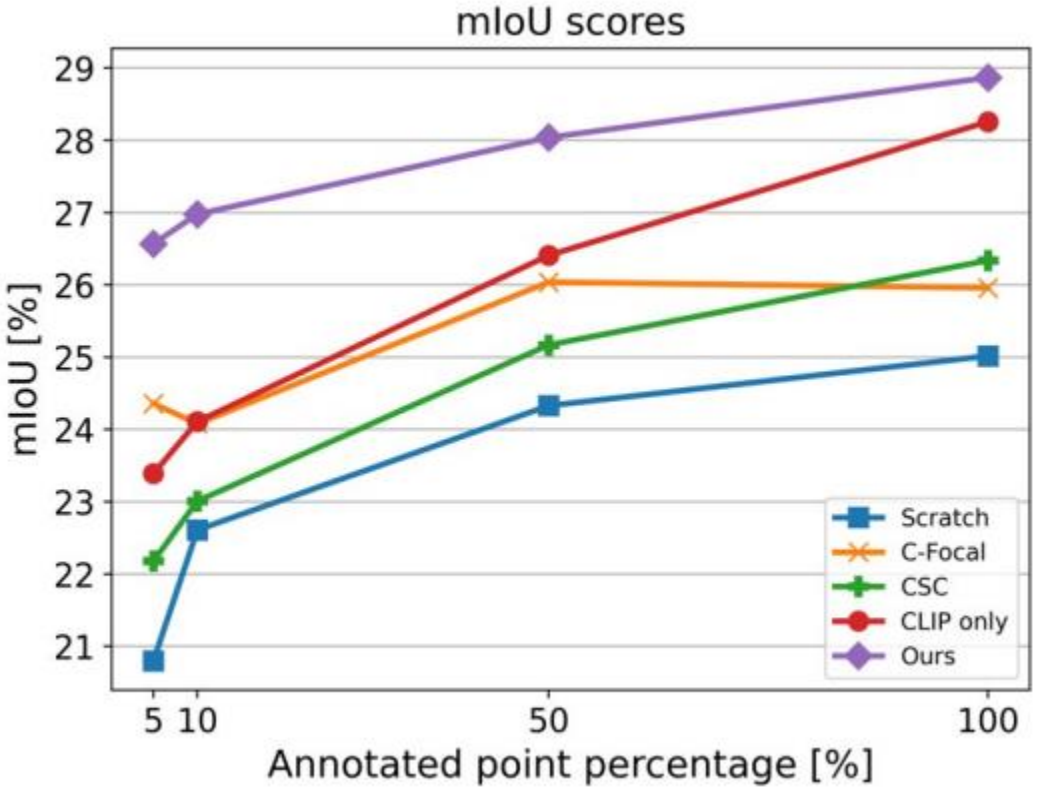


Ours

ScanNet200 [Rozenberszki et al. ‘22]

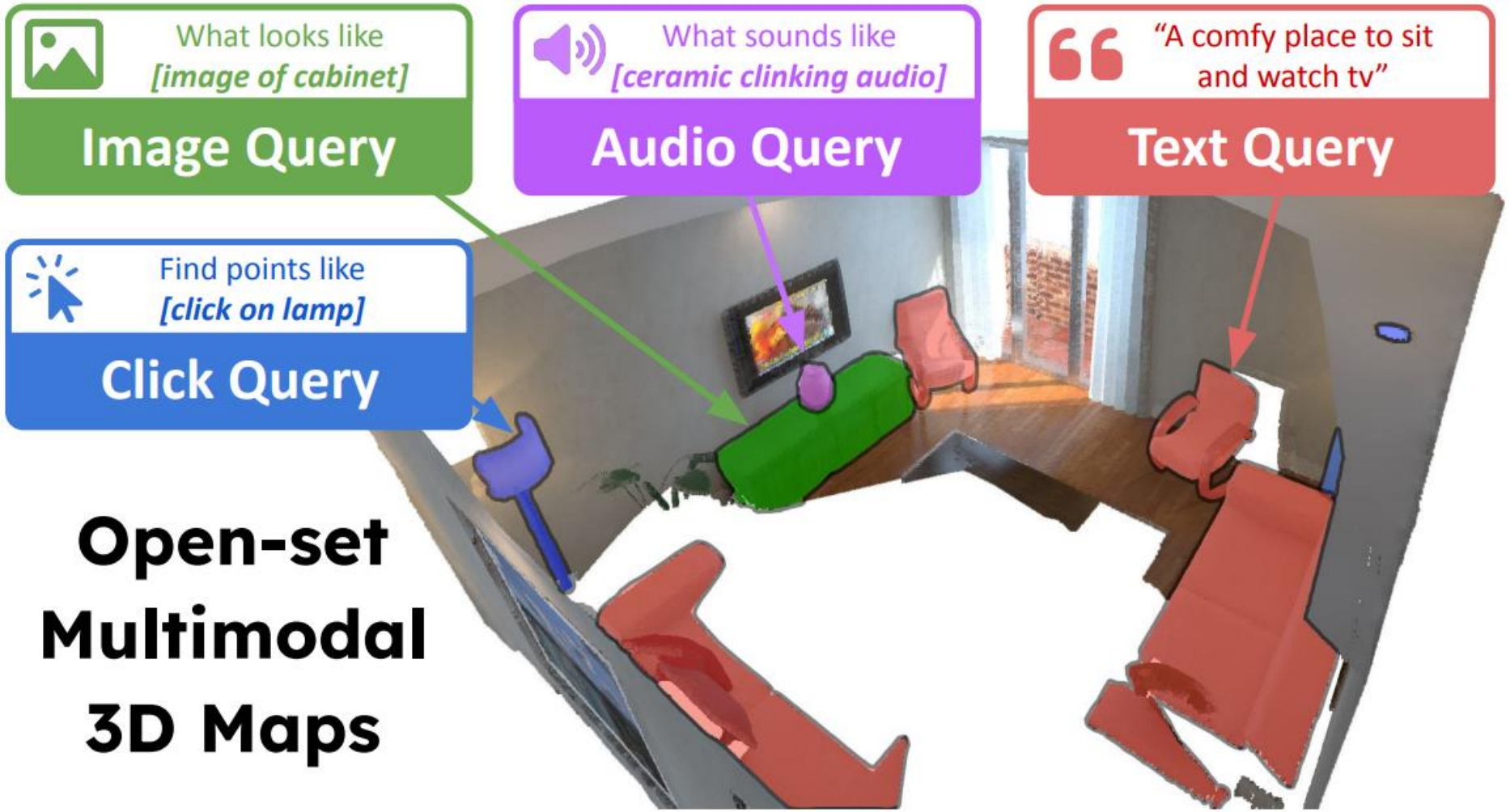
Language-Grounded 3D Learning

- Semantic segmentation on ScanNet200



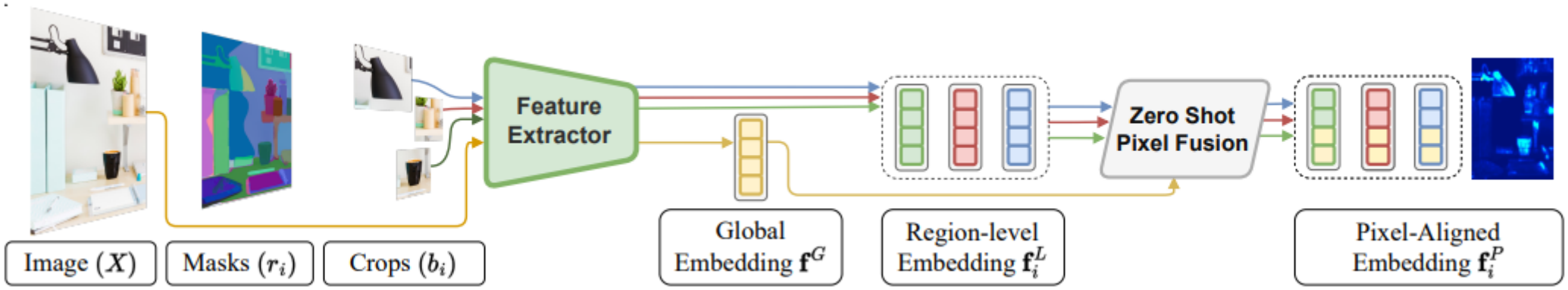
ScanNet200 [Rozenberszki et al. ‘22]

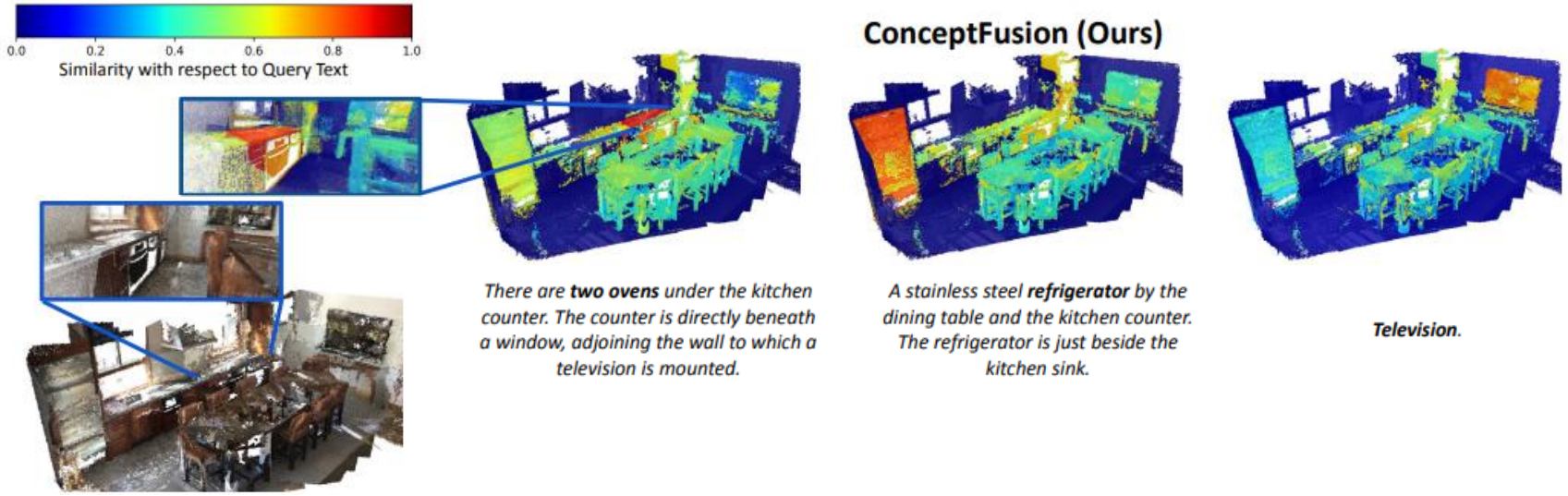
Multimodal 3D Mapping



ConceptFusion [Jatavallabhula et al. '23]

Multimodal 3D Mapping





ConceptFusion [Jatavallabhula et al. '23]

Next: How to distinguish objects?