HAN LAB
Hardware, AI and Neural-nets

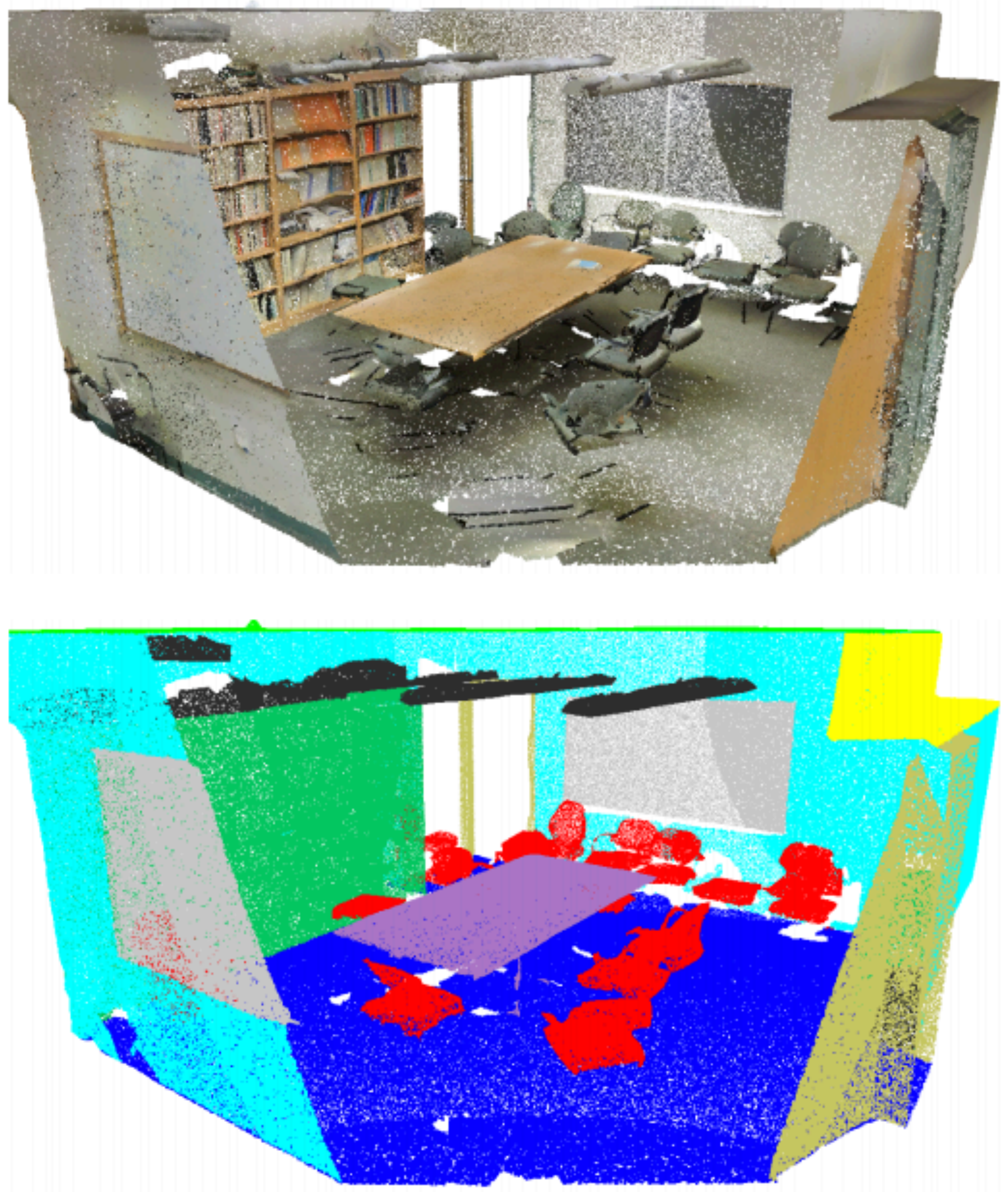
Point-Voxel CNN for Efficient 3D Deep Learning

Zhijian Liu*, **Haotian Tang***, **Yujun Lin**, and **Song Han**

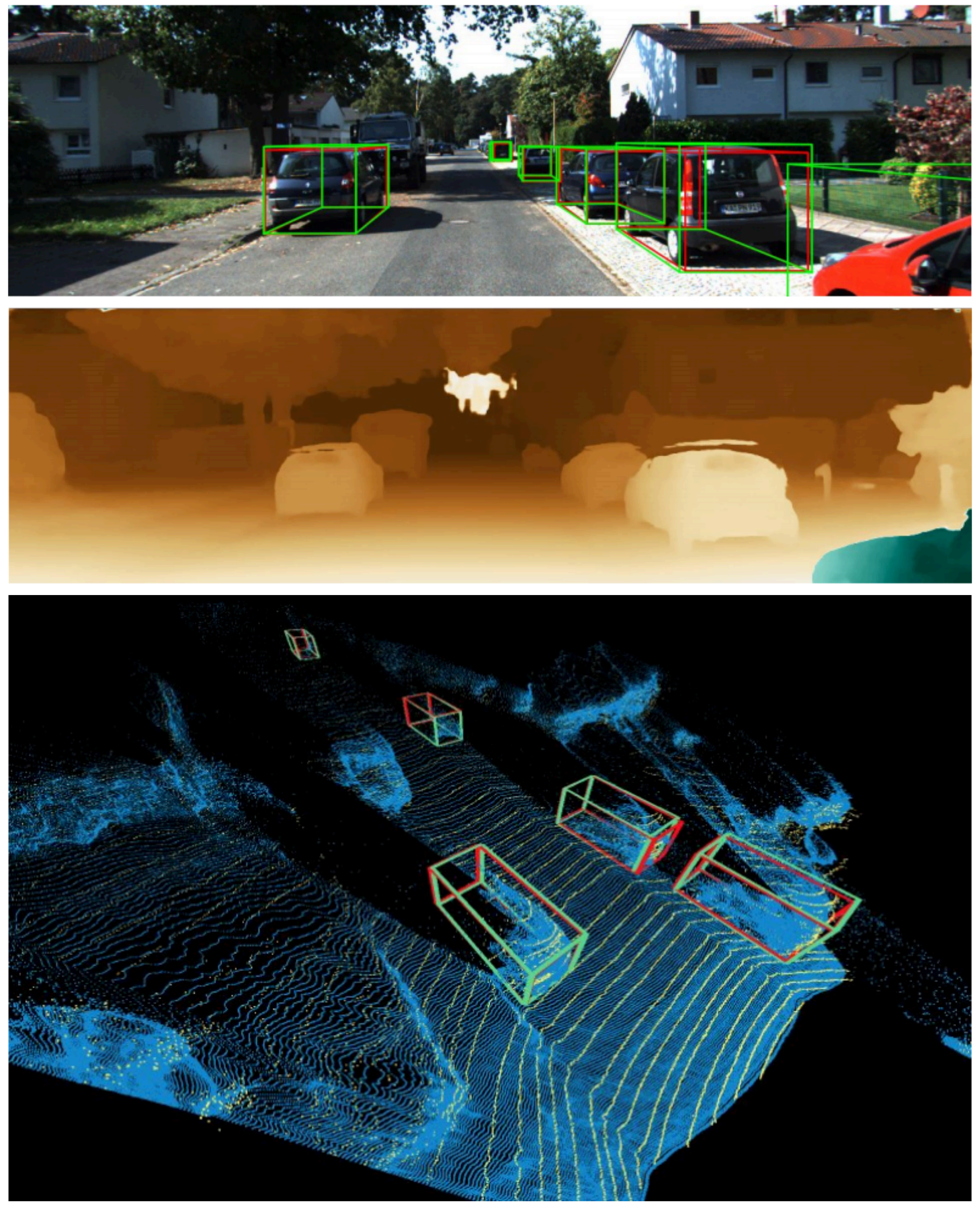
Project Page: **http://pvcnn.mit.edu/**

3D Deep Learning

3D Part Segmentation
(for Robotic Systems)



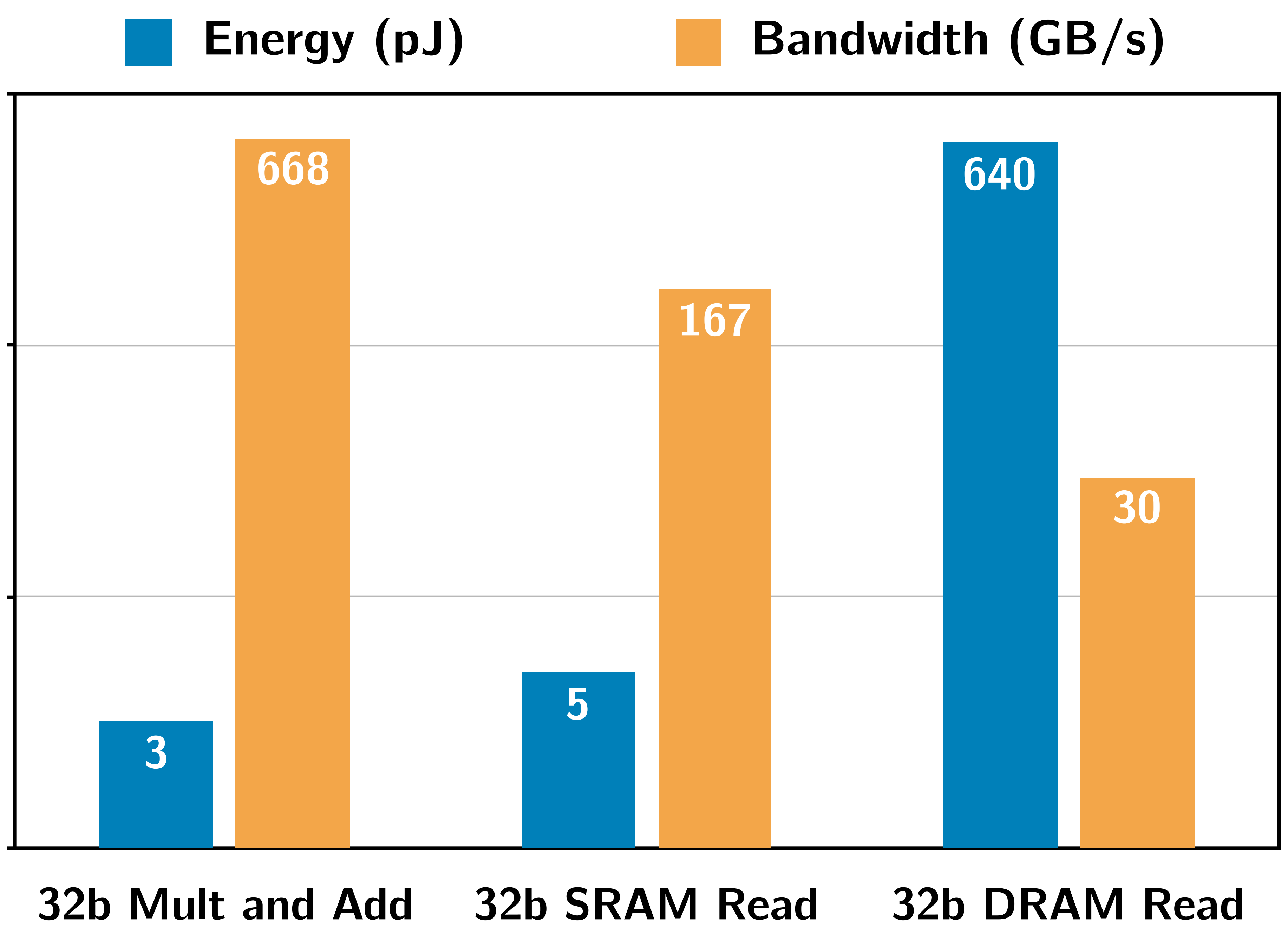
3D Semantic Segmentation
(for VR/AR Headsets)

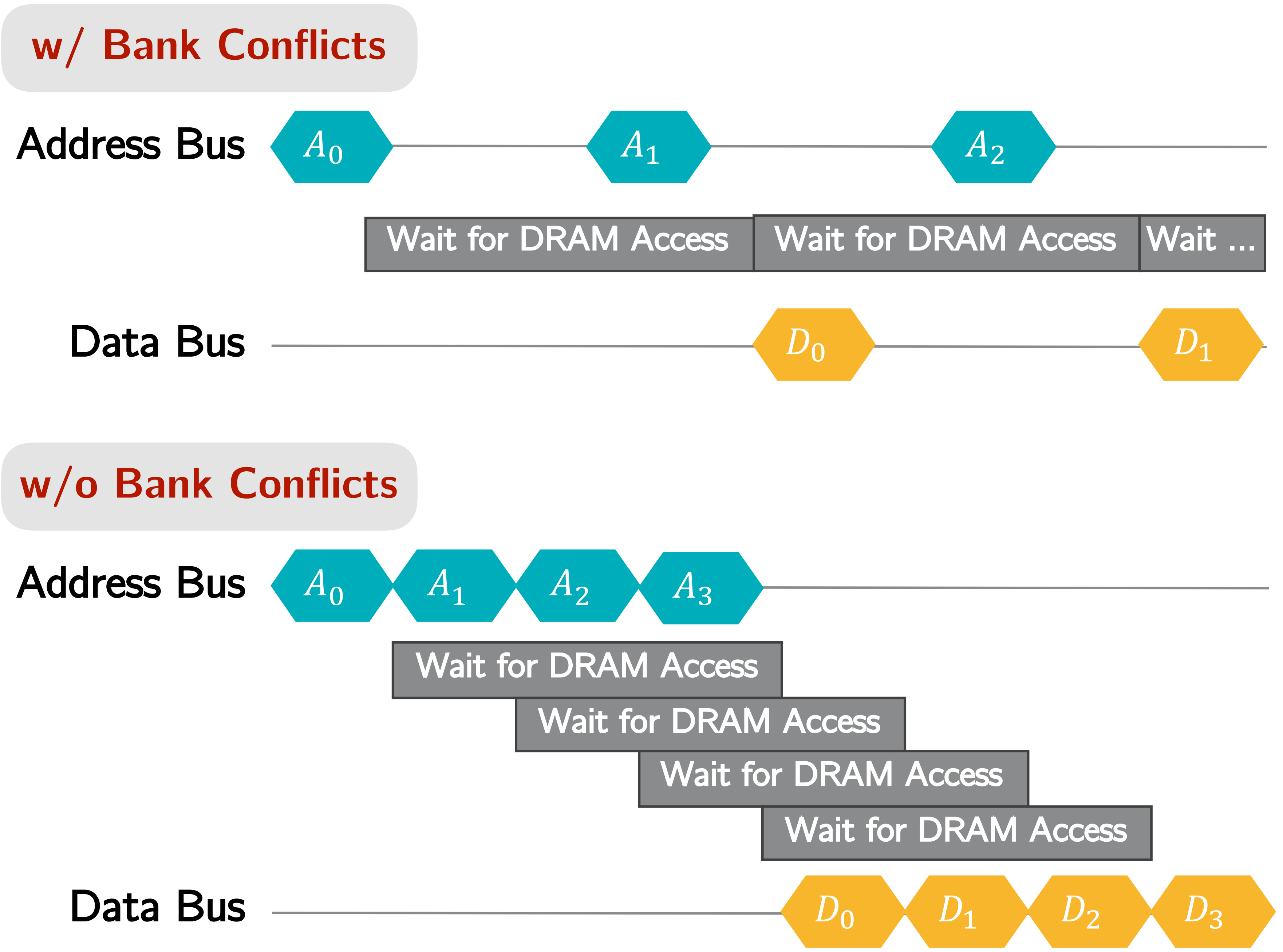


3D Object Detection
(for Self-Driving Cars)

3D deep learning has been used in various applications
on **resource-constrained** edge devices.

Efficient 3D Deep Learning



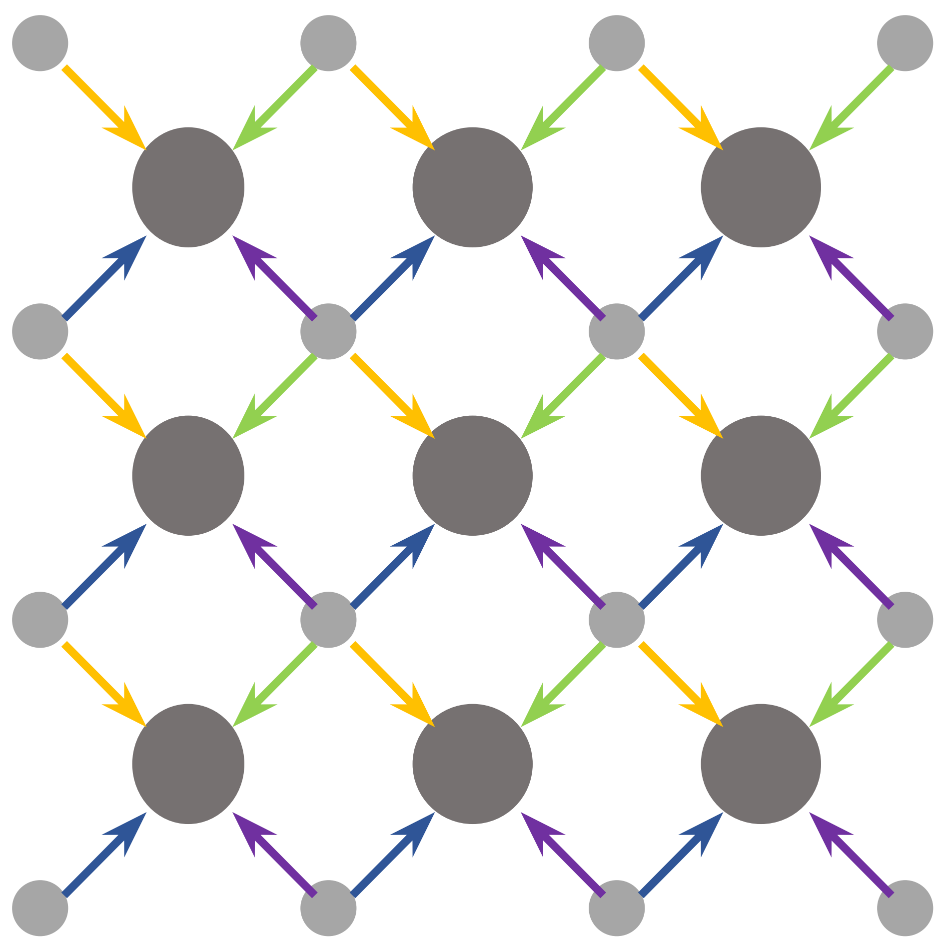


Off-chip DRAM access is much more expensive than arithmetic operation!

Random memory access is inefficient due to the potential bank conflicts!

Efficient 3D deep learning models should **have small memory footprints** and **avoid random memory access**.

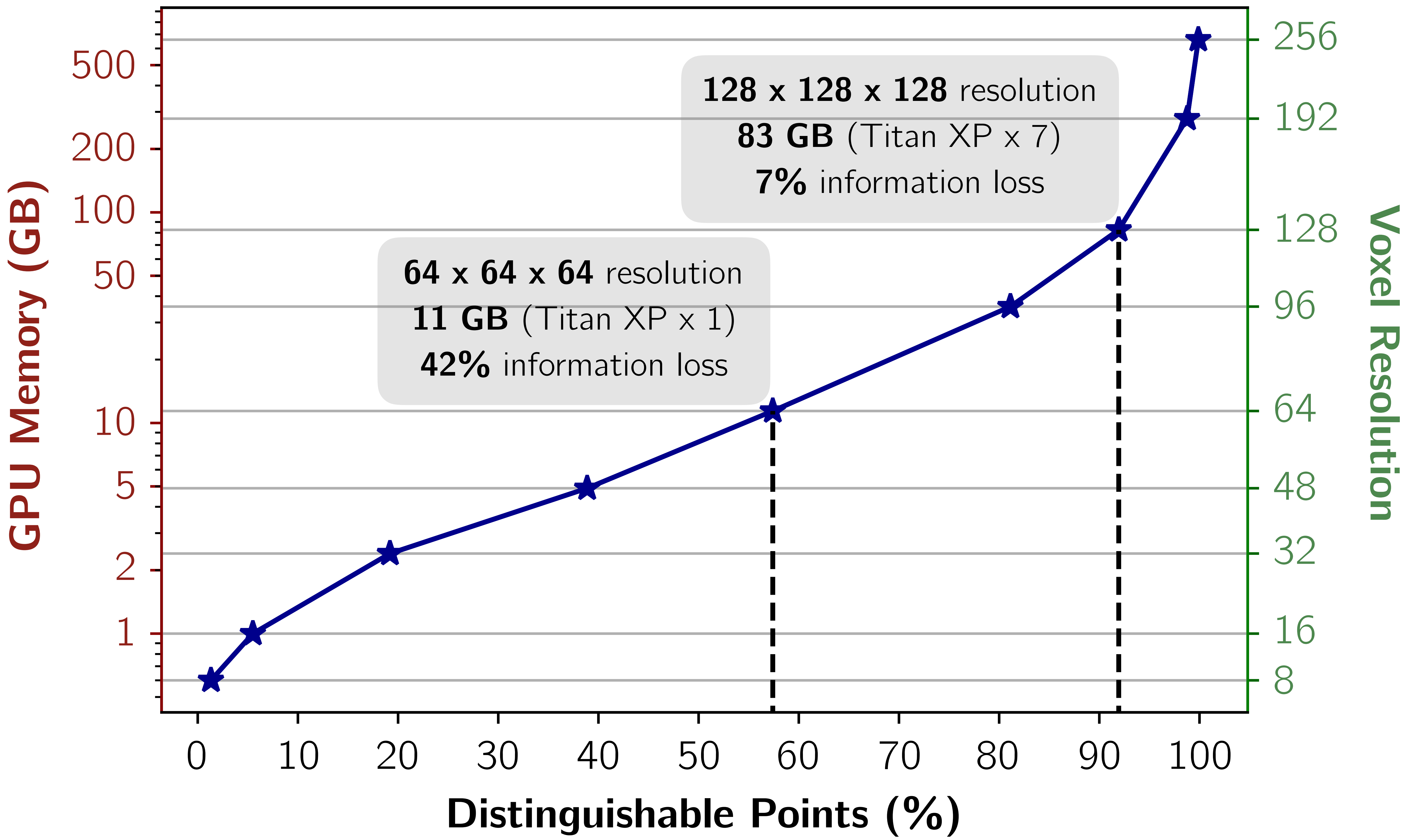
Voxel-Based Models: Cubically-Growing Memory



3D ShapeNets [CVPR'15]

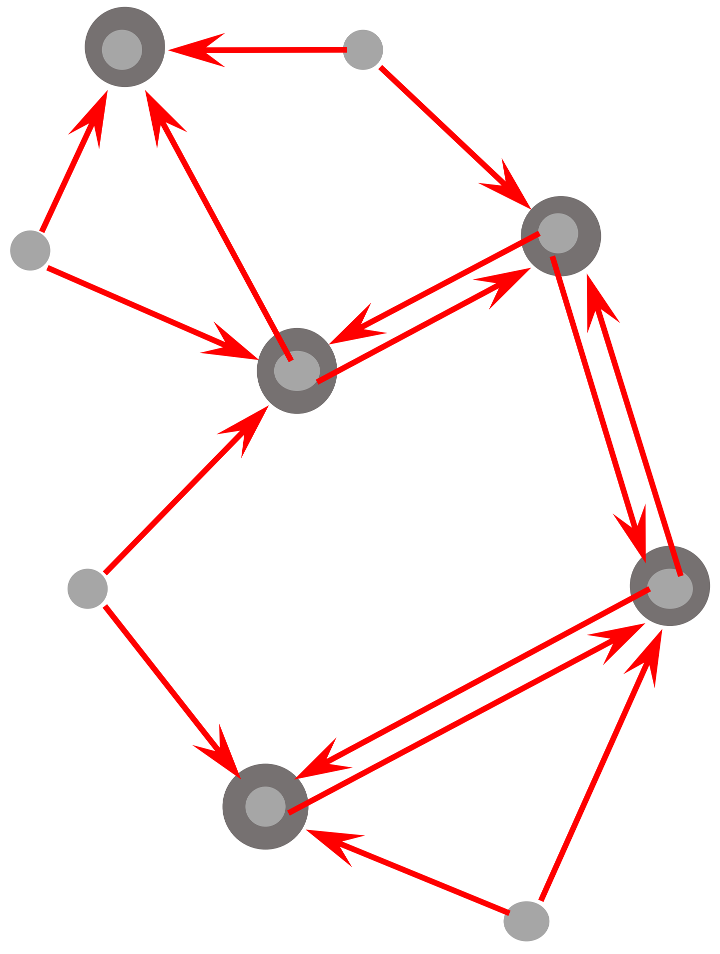
VoxNet [IROS'15]

3D U-Net [MICCAI'16]



Low resolutions lead to **significant information loss**.
High resolutions lead to **large memory consumption**.

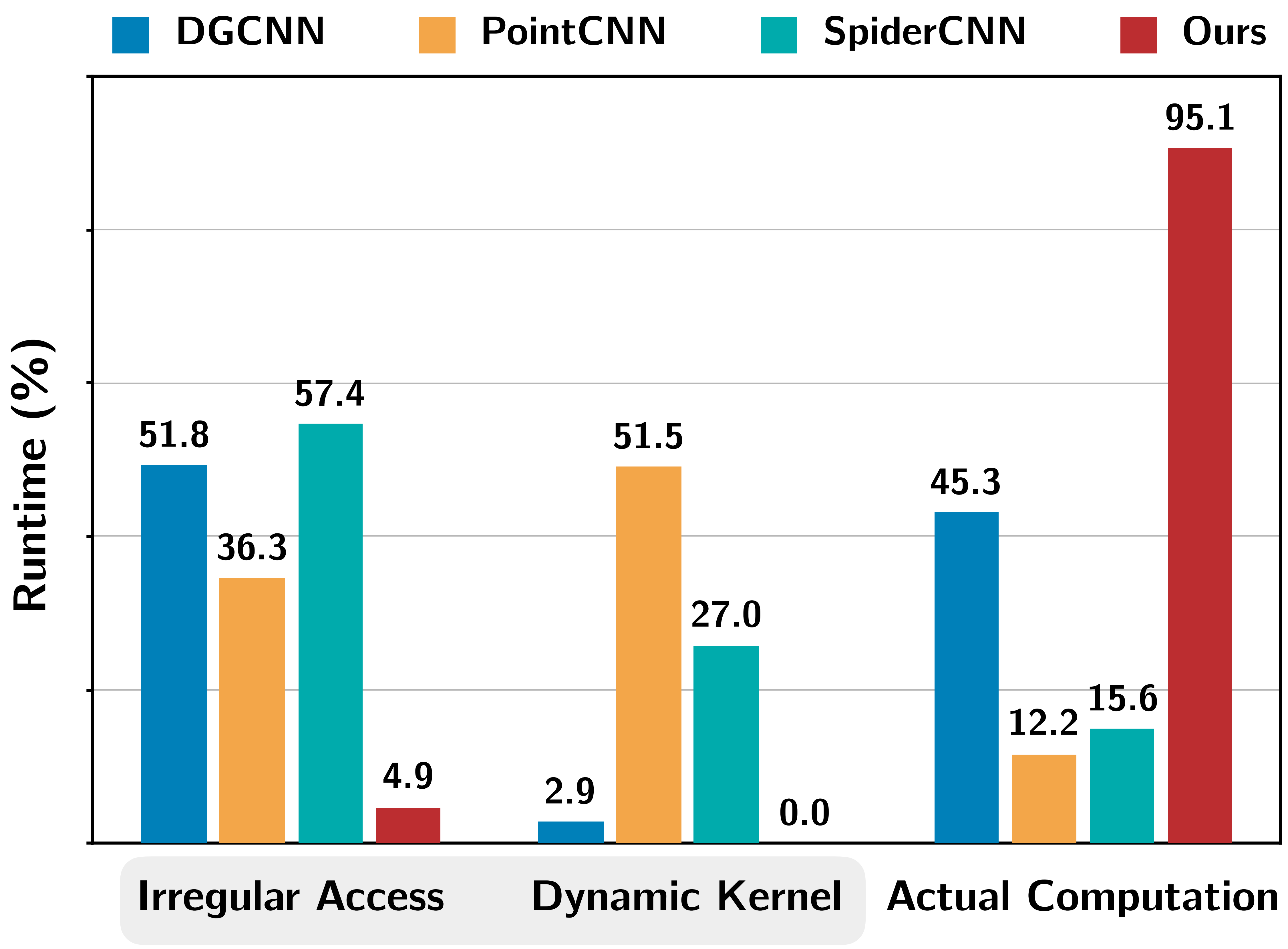
Point-Based Models: Sparsity Overheads



PointNet [CVPR'17]

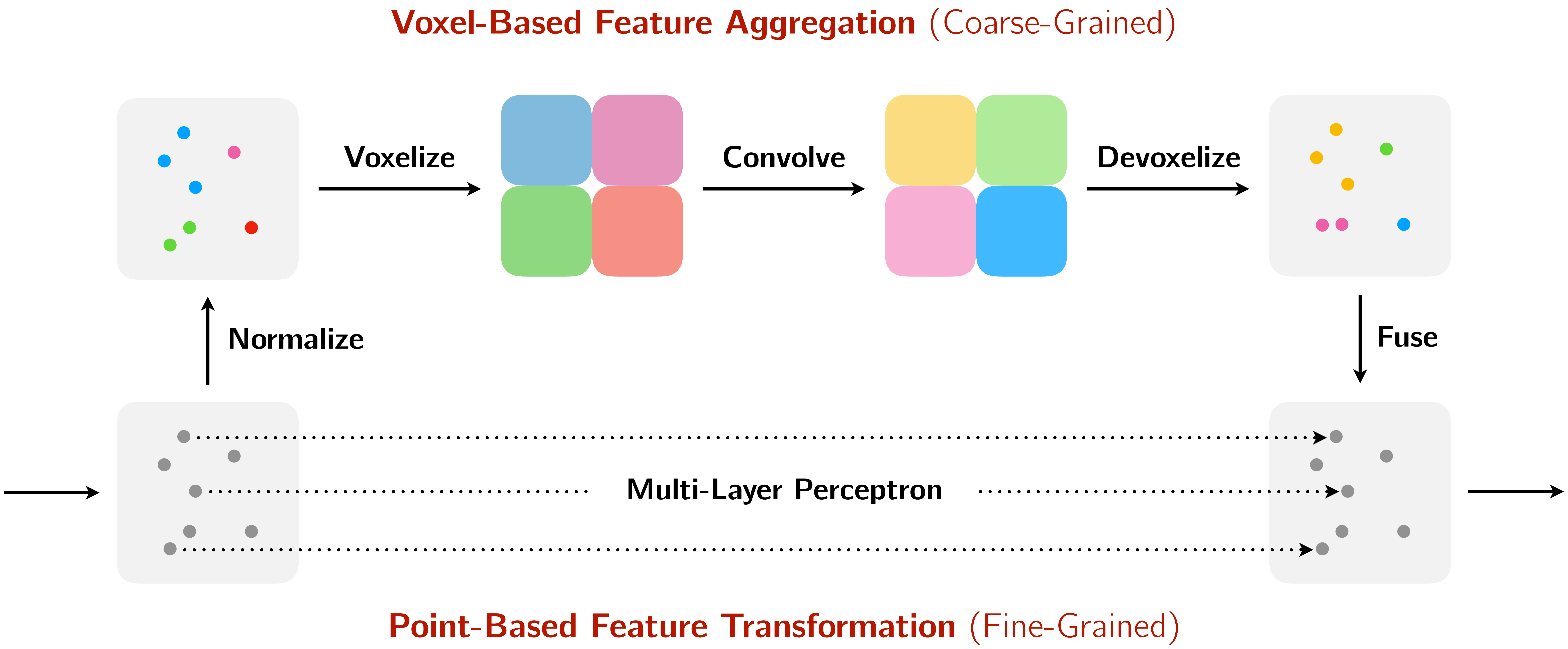
PointCNN [NeurIPS'18]

DGCNN [SIGGRAPH'19]



Up to **80% of the time** is wasted on **structuring the sparse data**,
not on the actual feature extraction.

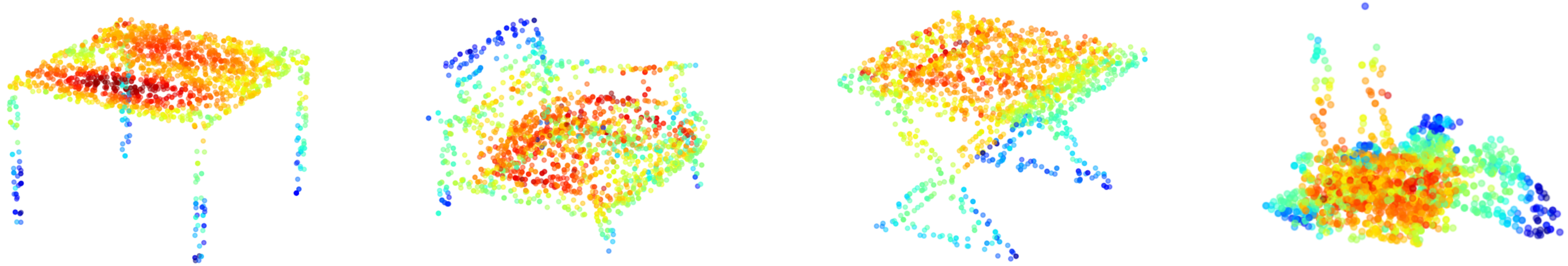
Point-Voxel Convolution (PVConv)



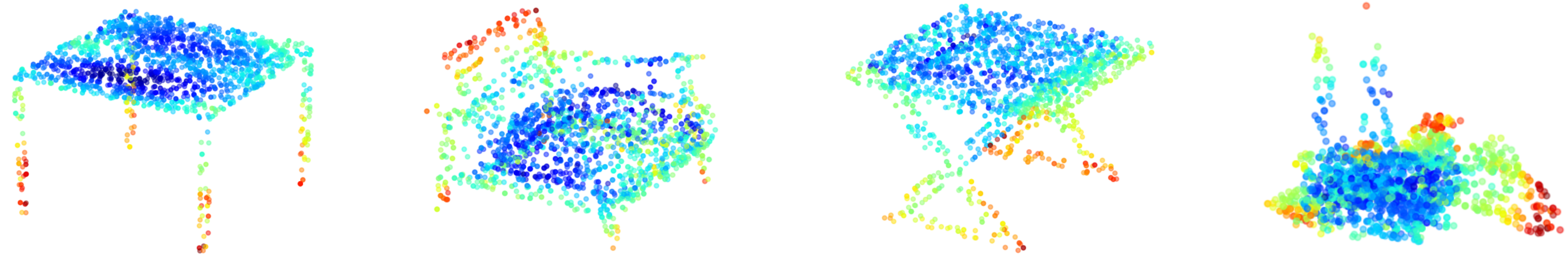
PVCNN combines the advantages of point-based models (**small memory footprint**) and voxel-based models (**regularity**).

Point-Voxel Convolution (PVConv)

Features from **Voxel-Based Branch**:

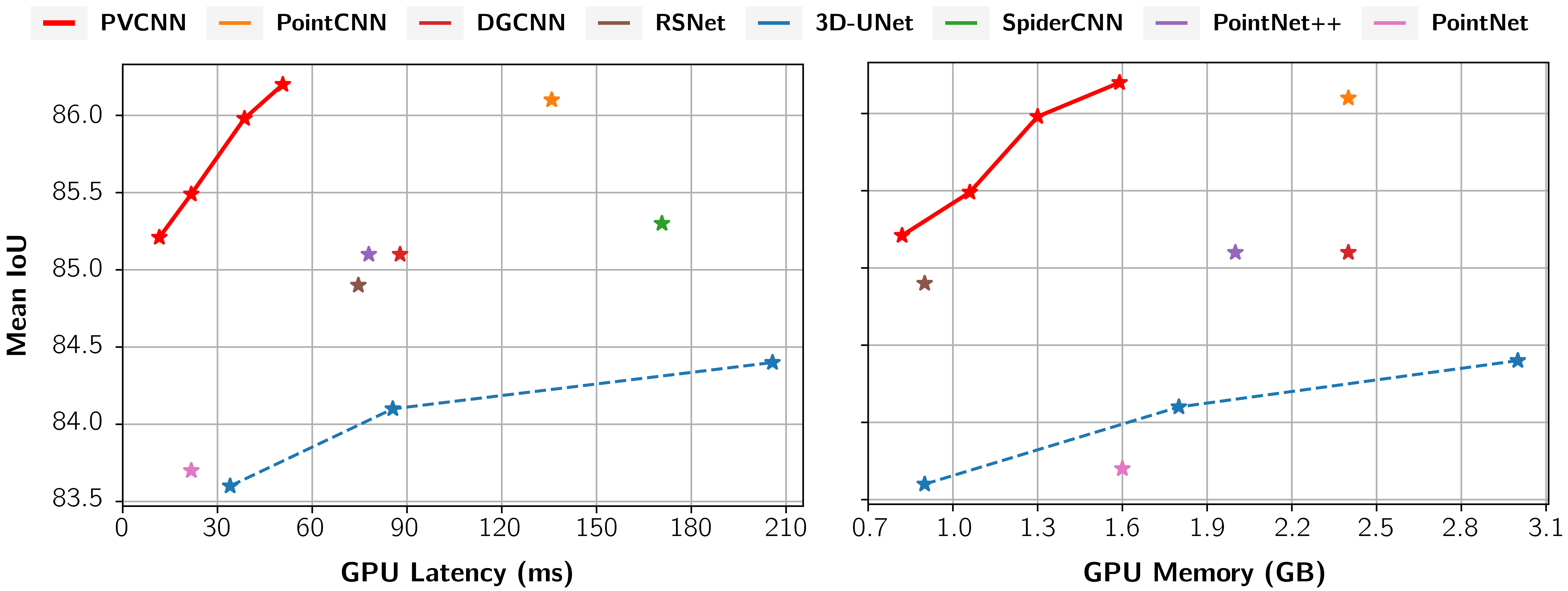


Features from **Point-Based Branch**:



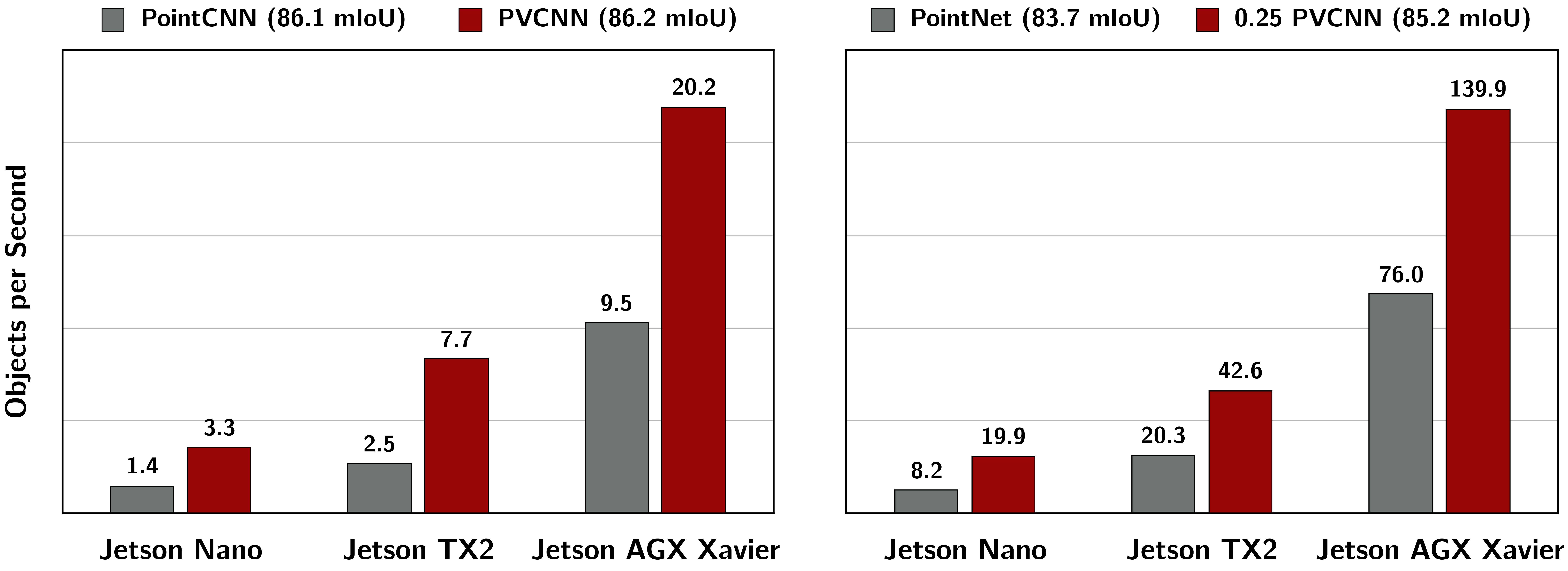
Voxel-based branch captures **large, contiguous** parts.
Point-based branch captures **isolated, discontinuous** details.

Results: 3D Part Segmentation (ShapeNet)



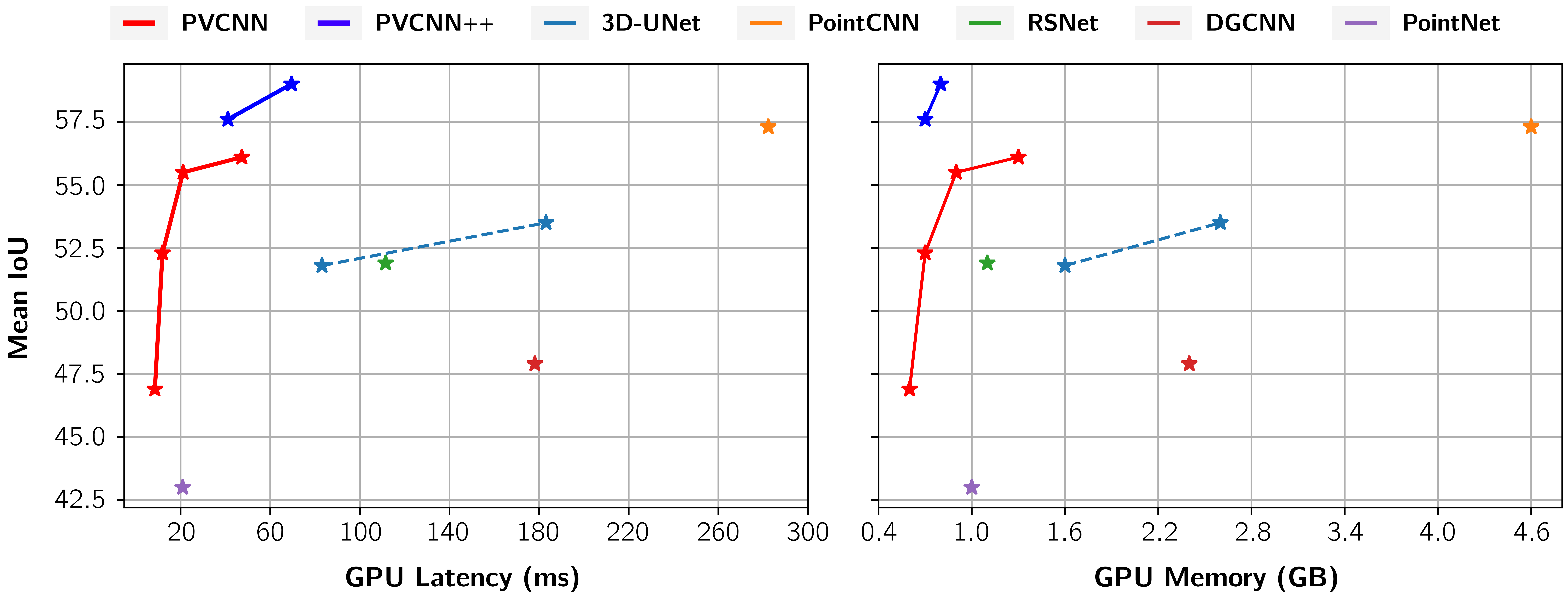
PVCNN outperforms PointCNN with **2.7x** measured speedup
and **1.5x** memory reduction (on a GTX 1080Ti GPU).

Results: 3D Part Segmentation (ShapeNet)



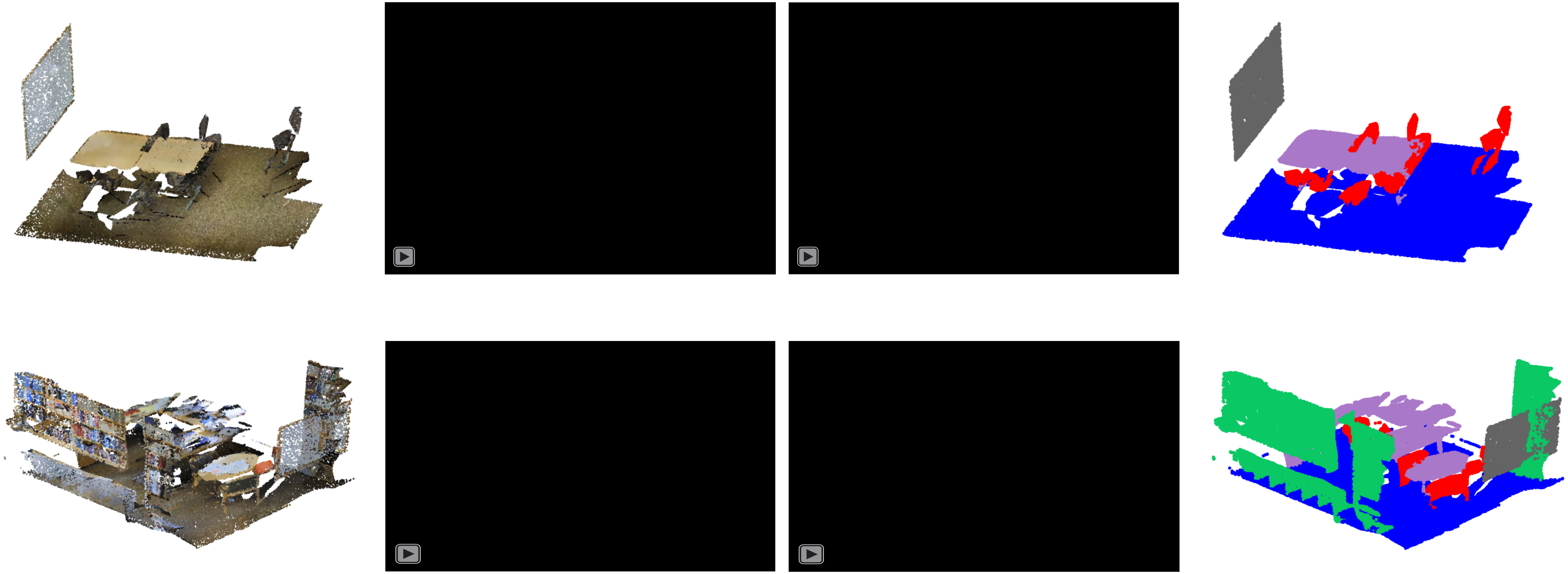
0.25 PVCNN runs with **real-time performance** (20 FPS) on the lightweight edge device (Jetson Nano).

Results: 3D Semantic Segmentation (S3DIS)



PVCNN++ outperforms PointCNN with **6.9x** measured speedup and **5.7x** memory reduction (on a GTX 1080Ti GPU).

Results: 3D Semantic Segmentation (S3DIS)



Input Scene **PointNet** **0.25 PVCNN** (Ours) **Ground Truth**

0.25 PVCNN outperforms PointNet with **1.8x** measured speedup and **1.4x** memory reduction (on a GTX 1080Ti GPU).

Results: 3D Object Detection (KITTI)

	Efficiency		Car			Pedestrian			Cyclist		
	Latency (GPU)	Memory (GPU)	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
F-PointNet++	105.2 ms	2.0 GB	83.8	70.9	63.7	70.0	61.3	53.6	77.2	56.5	53.4
PVCNN (efficient)	**58.9 ms** **(1.8x)**	**1.4 GB** **(1.4x)**	**84.2** **(+0.4)**	71.1 (+0.2)	63.6 (-0.1)	69.2 (-0.8)	60.3 (-1.0)	52.5 (-1.1)	78.7 (+1.5)	57.8 (+1.3)	54.2 (+1.2)
PVCNN (complete)	69.6 ms (1.5x)	**1.4 GB** **(1.4x)**	84.0 (+0.2)	**71.5** **(+0.6)**	**63.8** **(+0.1)**	**73.2** **(+3.2)**	**64.7** **(+3.4)**	**56.8** **(+3.2)**	**81.4** **(+4.2)**	**60.0** **(+3.5)**	**56.3** **(+2.9)**

PVCNN outperforms F-PointNet++ by **2.4%** mAP with **1.5x** measured speedup and **1.4x** memory reduction.

Results: 3D Object Detection (KITTI)

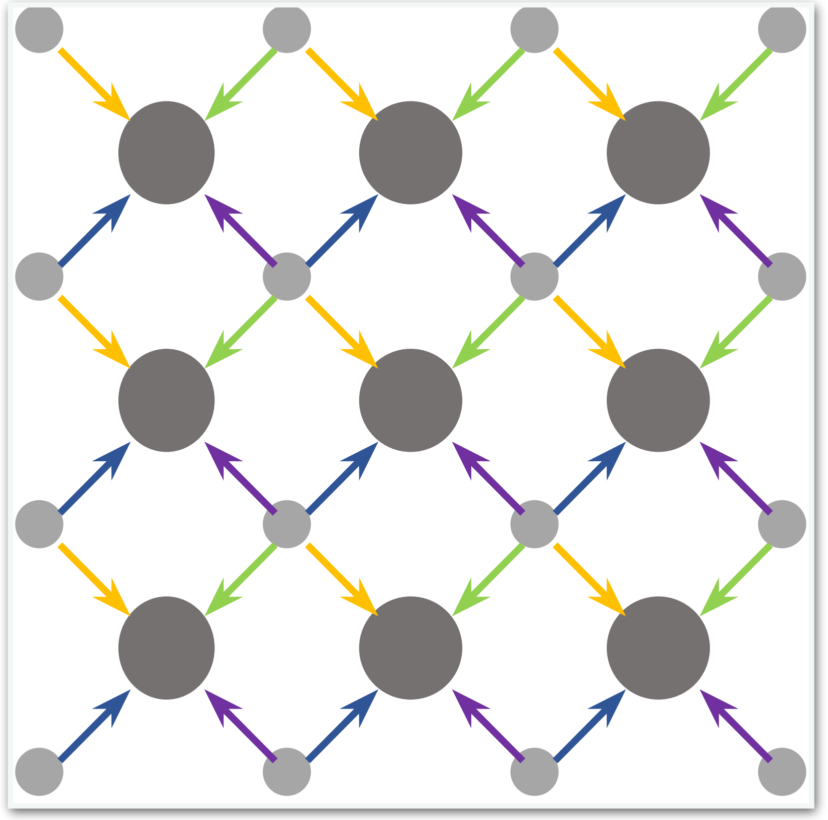
F-PointNet++

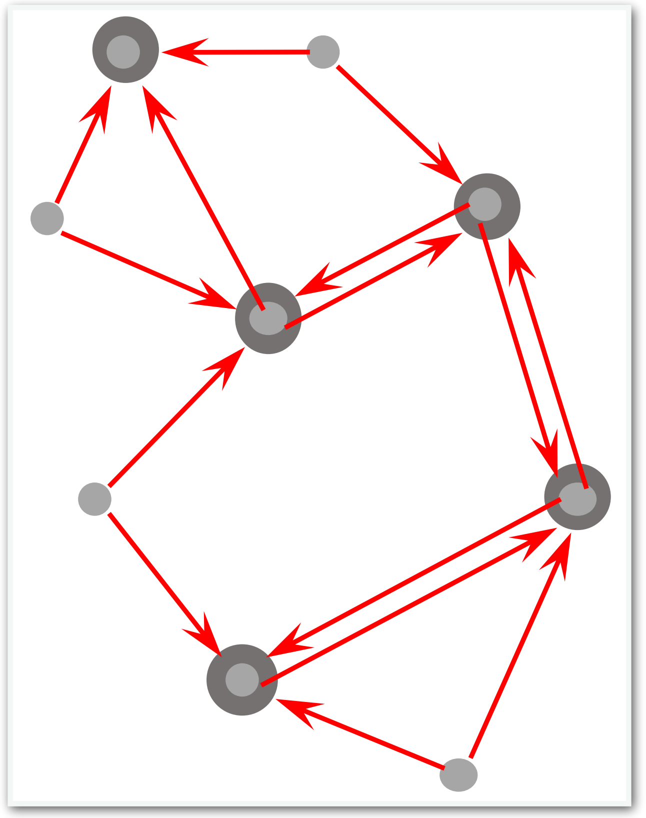
PVCNN (Ours)

PVCNN outperforms F-PointNet++ by **2.4%** mAP with
1.5x measured speedup and **1.4x** memory reduction.

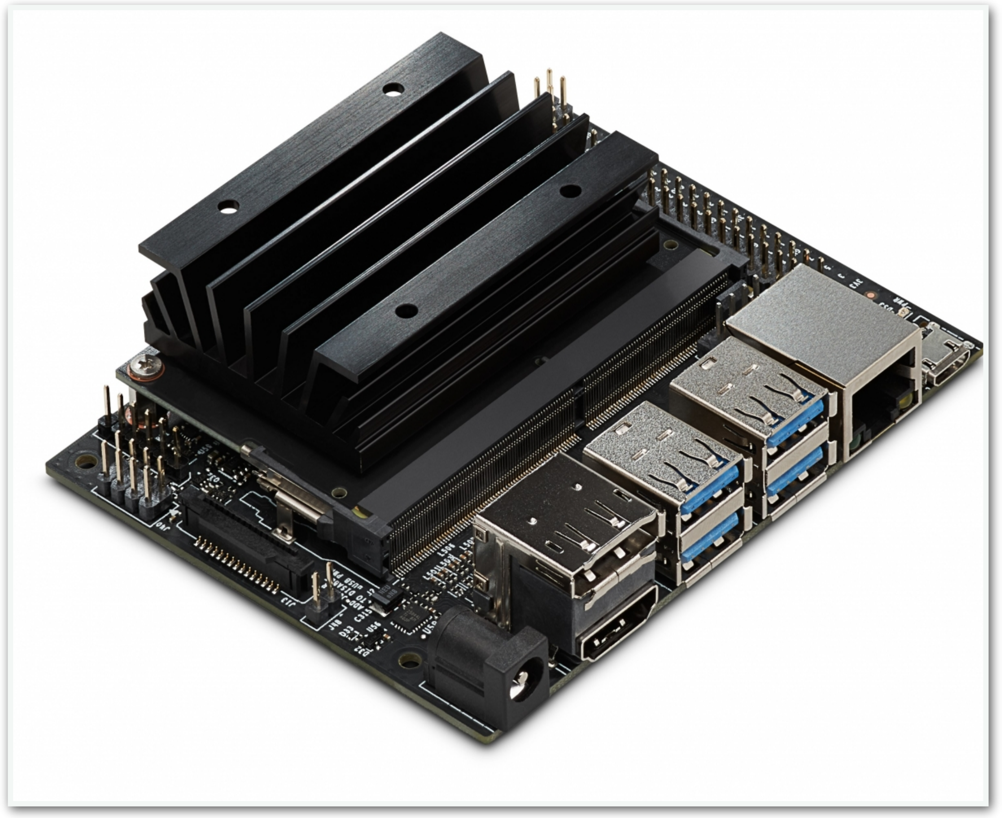
Point-Voxel CNN for Efficient 3D Deep Learning

Bottleneck Analysis





Hardware-Efficient Primitive



Project Page: **http://pvcnn.mit.edu/**