VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking

Yukang Chen[1], Jianhui Liu[2], Xiangyu Zhang[3], Xiaojuan Qi[2], Jiaya Jia[1]

[1] Chinese University of Hong Kong, [2] University of Hong Kong, [3] MEGVII Technology

https://github.com/dvlab-research/VoxelNeXt

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- Highlight

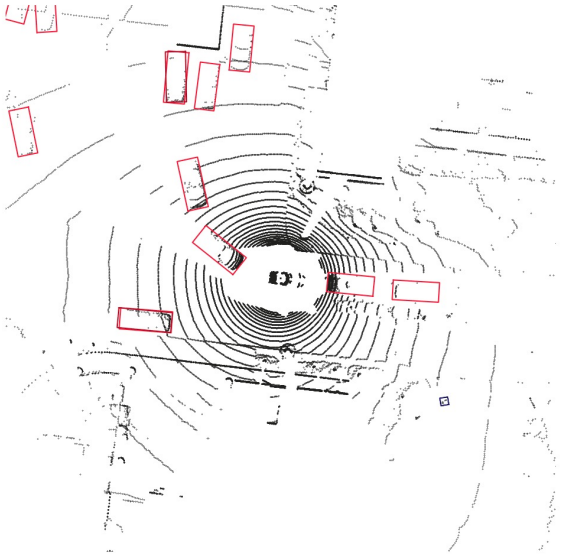
- **Fully sparse**, Fully Voxel-based, LiDAR Detector
- (**1st** on nuScenes LiDAR Tracking Leaderboard)
- **High efficiency**: 64 ms per frame on nuScenes

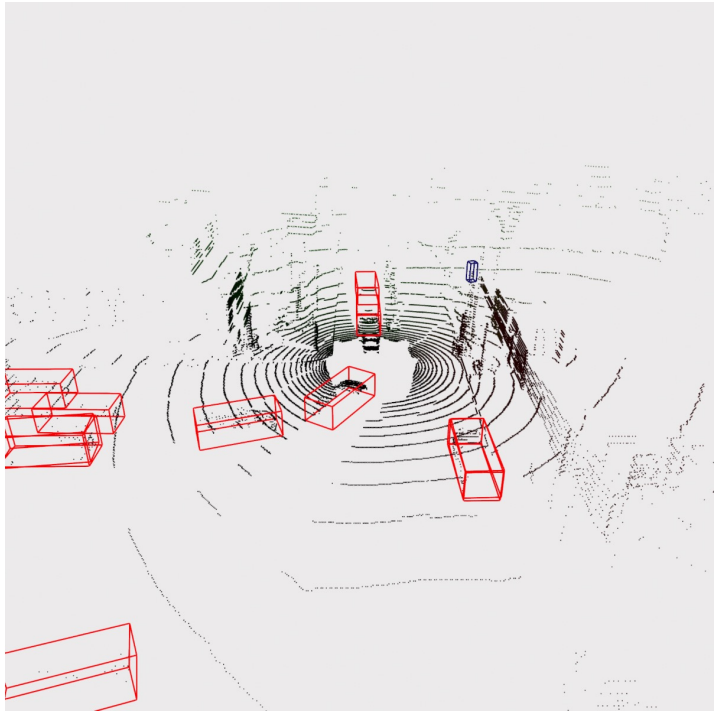
No need for conventional designs

Sparse-to-Dense Convertion ✗

Anchors *Centers* *NMS*

RPN *Dense head* *RoI Pooling*

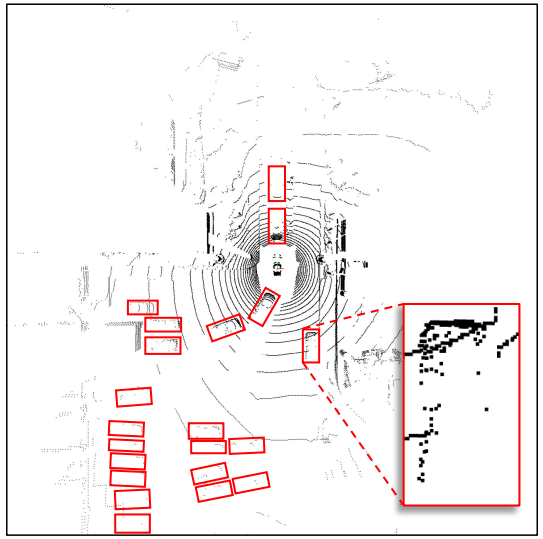




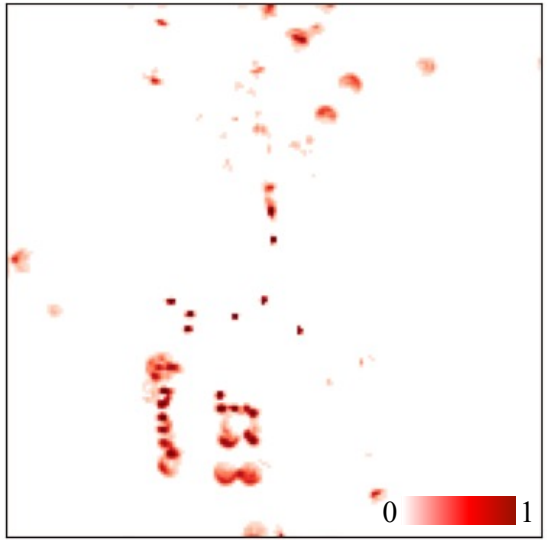
	Method					
	Date	Name	Modalities	Map data	External data	AMOTA
			Lidar	All	All	
>	2023-03-09	FocalFormer3D	Lidar	no	no	0.715
>	2022-11-11	VoxelNeXt	Lidar	no	no	0.710
>	2022-11-04	L-ByteTrack	Lidar	no	no	0.701
>	2022-08-02	Minkowski Tracker	Lidar	no	no	0.698

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

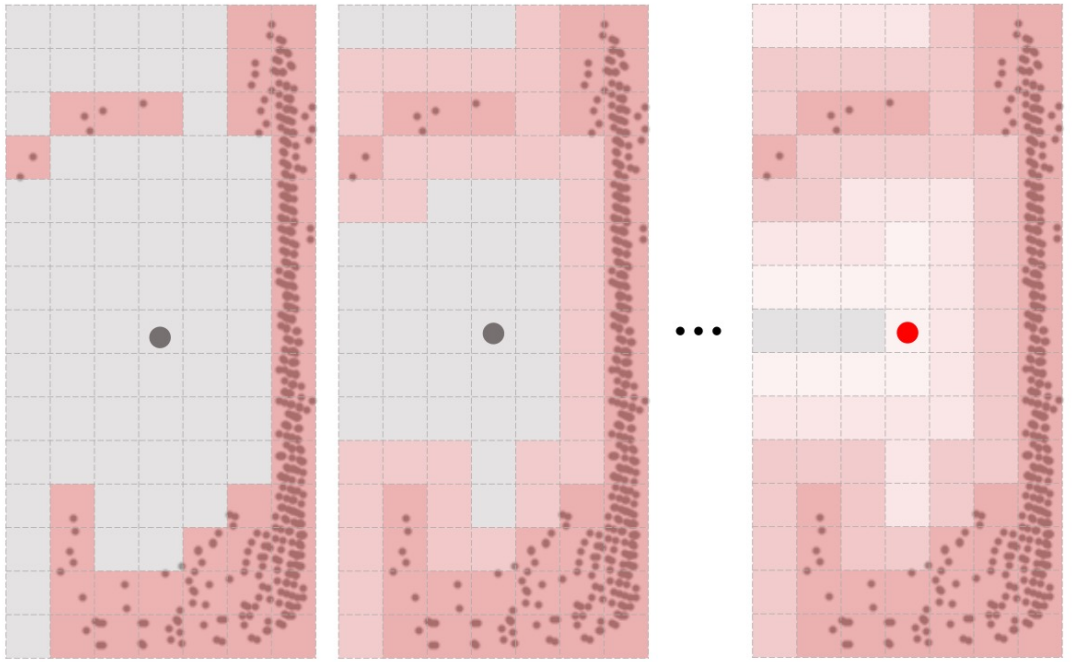
- **Why fully sparse?**



Point cloud &
Ground-truth boxes



Heatmaps from CenterPoint



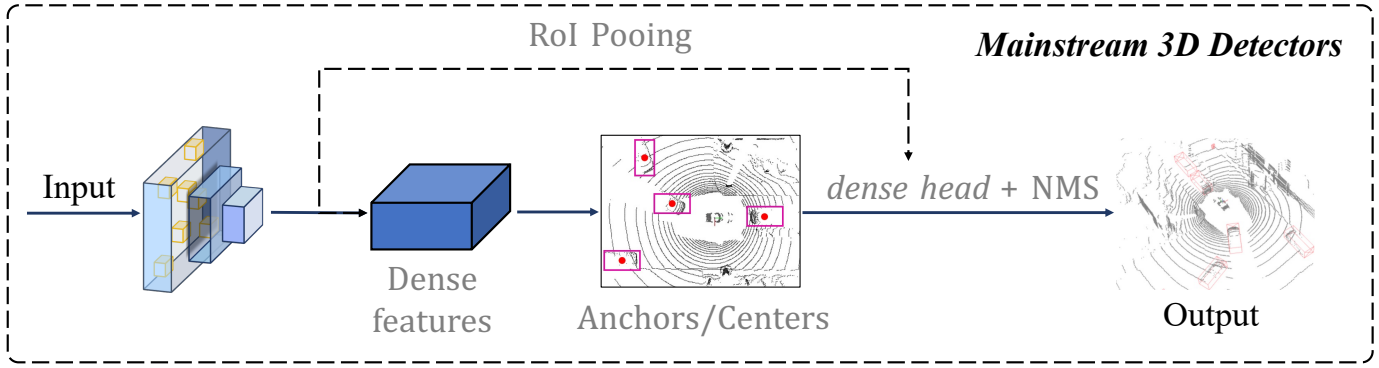
Most areas are empty and contain no point clouds.
The whole BEV dense features are involved in computation.

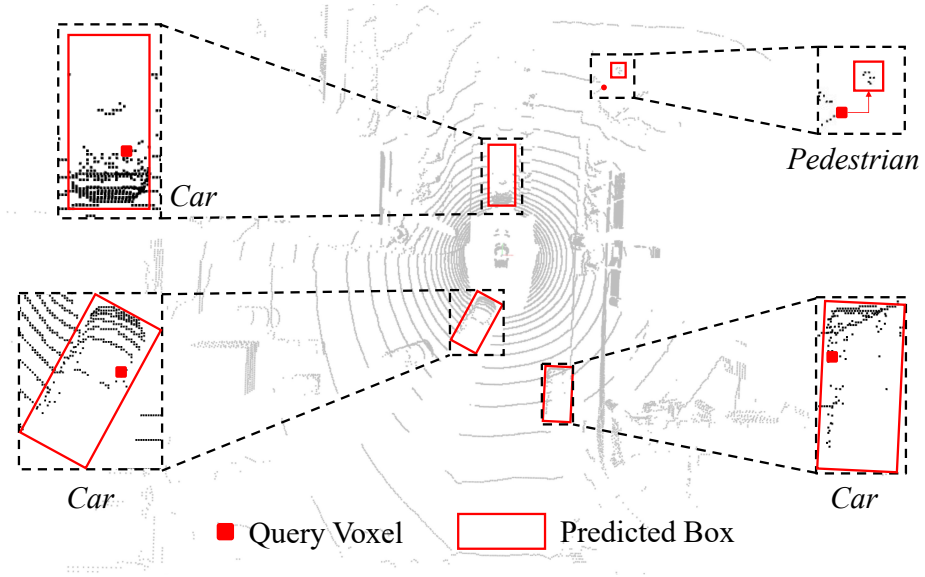
Point clouds only exist on object boundary [1].
Dense convs are required to resolve the center-missing issue.

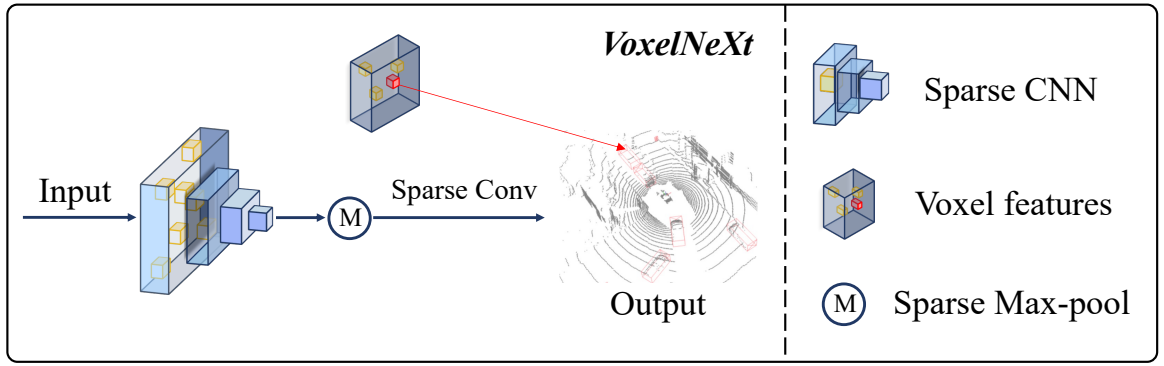
[1] Lue Fan, Feng Wang, Naiyan Wang, Zhaoxiang Zhang, Fully Sparse 3D Object Detection, NeurIPS 2022

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Our framework**



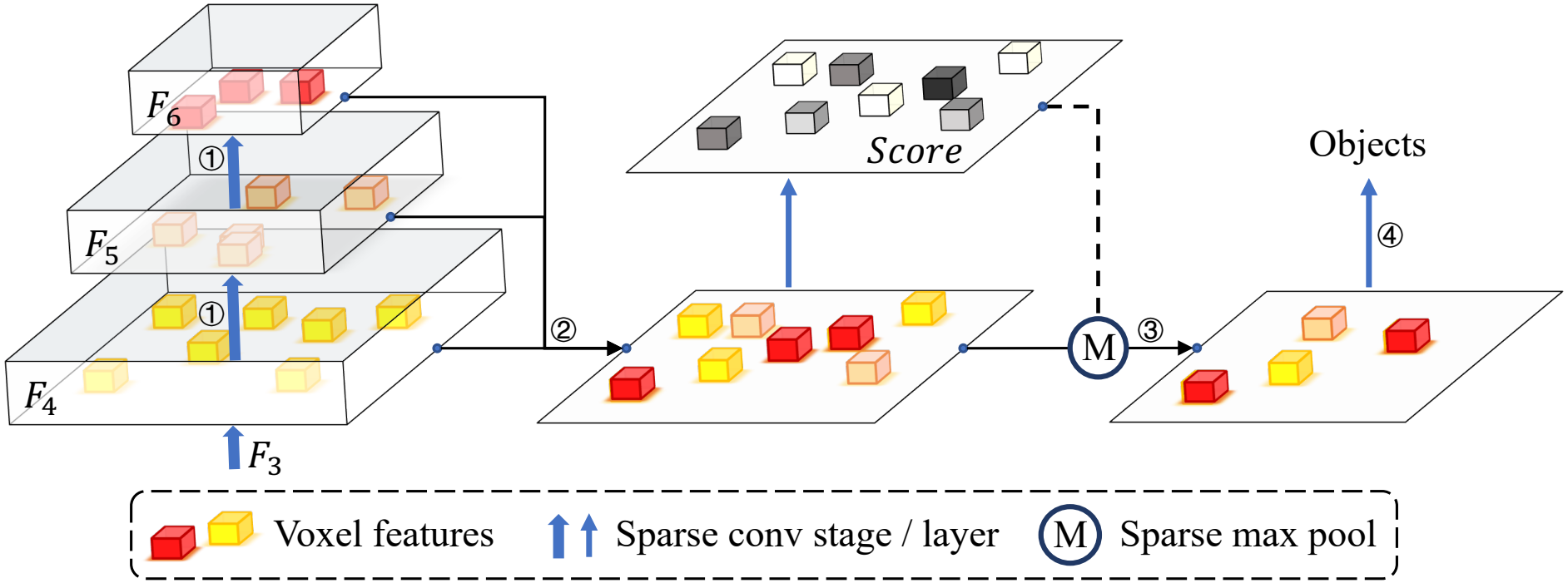




Method	mAP	NDS	FLOPs		Latency
			Sparse CNN	Head	
SECOND [46]	50.6	62.3	62.9 G	64.1 G	**64 ms**
CenterPoint [50]	58.6	66.2	62.9 G	123.7 G	96 ms
VoxelNeXt	**60.0**	**67.1**	**33.6 G**	**5.1 G**	66 ms

Core idea: Box prediction directly from sparse voxel features.

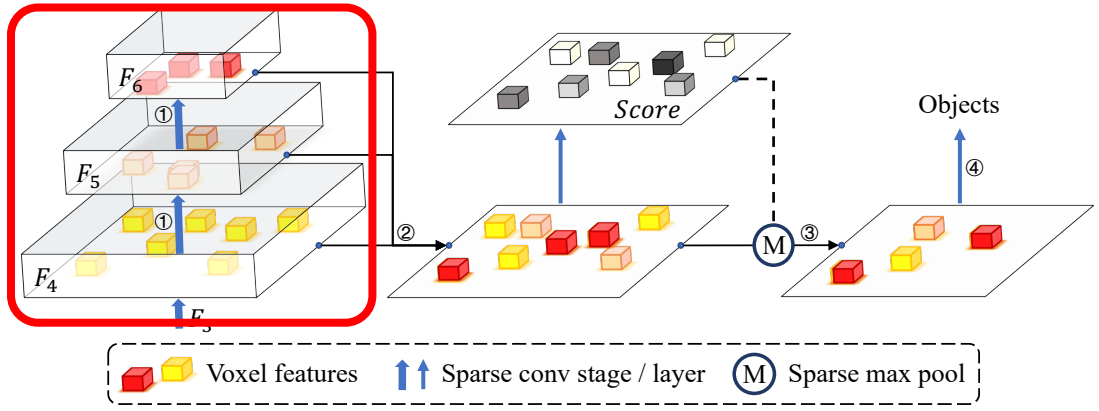
- **Detailed designs**

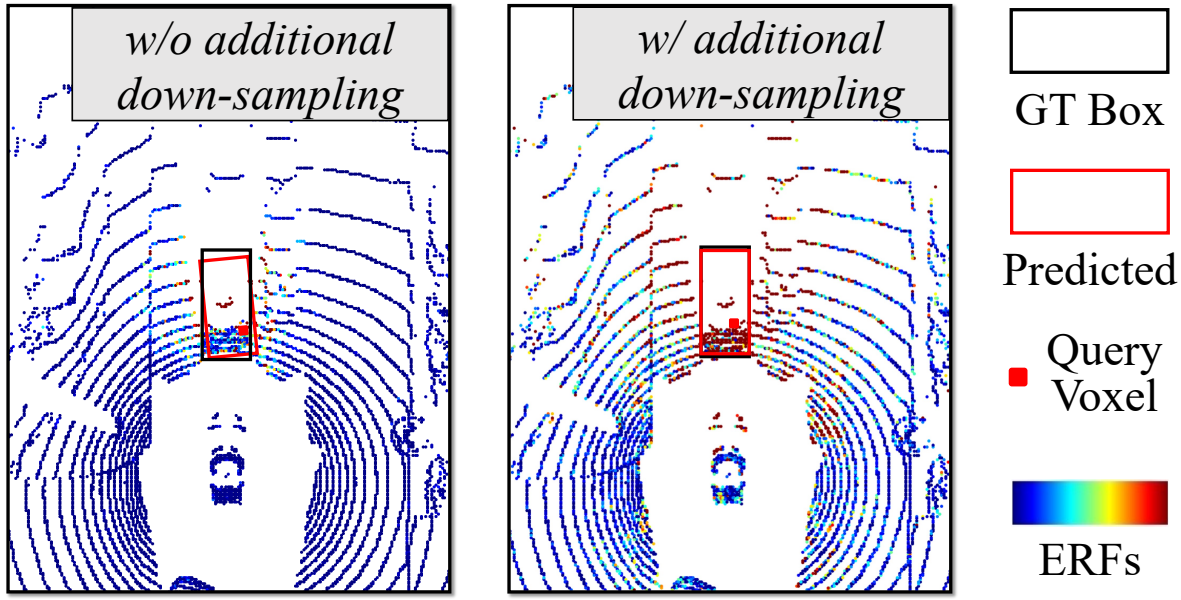


(1) Two additional down-samplings [For performance]
(2) Sparse Height Compression [For efficiency]
(3) Spatially Voxel Pruning [For efficiency]
(4) Sparse Max Pooling [For efficiency]

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Detail 1: Two additional down-samplings**





Method	*Strides*	Latency	mAP	NDS	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
CenterPoint	{2, 4, 8}	96 ms	55.6	63.2	83.5	54.9	67.5	30.6	16.3	83.3	52.7	34.5	65.6	66.5
D_3	{2, 4, 8}	56 ms	46.7$_{\downarrow 8.9}$	56.2	75.3	41.3	38.3	10.5	14.9	82.0	47.7	28.3	63.6	64.2
$D_3^{5\times5\times5}$	{2, 4, 8}	225 ms	51.6$_{\uparrow 4.9}$	60.4	80.0	49.2	56.8	16.8	16.5	83.5	50.2	30.9	64.8	67.7
D_4	{2, 4, 8, 16}	62 ms	52.3$_{\uparrow 5.6}$	61.2	80.0	50.0	61.2	23.1	16.9	82.5	49.0	31.8	63.9	64.8
D_5	{2, 4, 8, 16, 32}	66 ms	**56.5**$_{\uparrow 9.5}$	**64.5**	83.0	54.0	67.4	32.9	20.0	84.1	52.7	35.7	66.6	65.3

Simple to implement & significantly effective.

- **Detail 2: Sparse Height Compression**

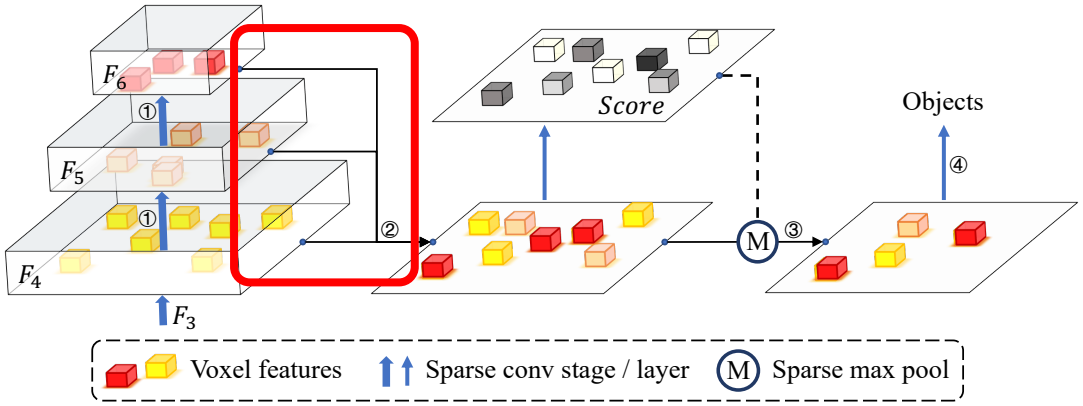


Table 5. Ablations on 2D or 3D sparse CNN in VoxelNeXt. sparse height Compression is used to connect 3D backbone and 2D head.

Method	Backbone	Head	Latency	mAP	NDS
-	3D	3D	92 ms	56.3	63.4
VoxelNeXt	3D	2D	66 ms	**56.2**	**64.3**
VoxelNeXt-2D	2D	2D	**61 ms**	53.4	62.6

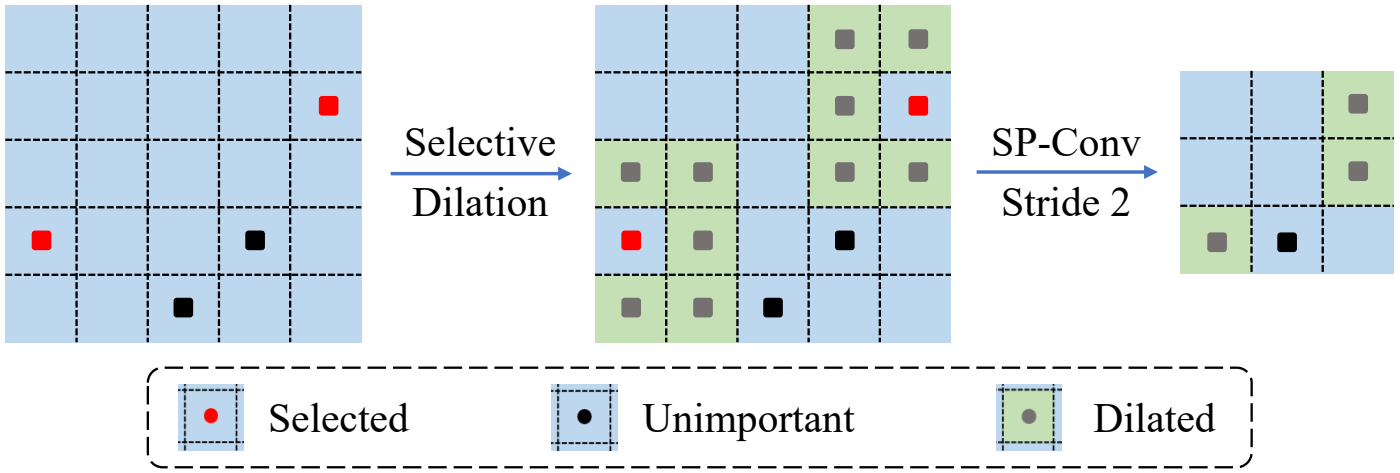
- **Put all 3D voxels onto BEV ground.**

From (x, y, z) → (x, y), no sparse-to-dense conversion.

- Enable 2D sparse head prediction, reduce voxel numbers from prediction.

Efficiency: 92 ms → 66 ms

- **Detail 3: Spatially Voxel Pruning**



$$M(F) = \text{Sigmoid}(G(F)), \text{where } G(F) = \frac{1}{C} \cdot \sum_{c=1}^{C} |F_c|$$

Directly remove voxels with small feature magnitudes.

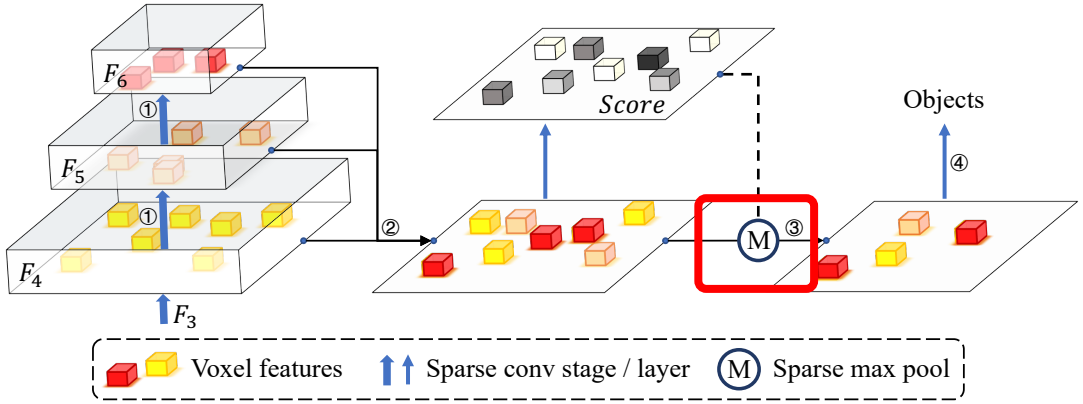
Table 3. Effects of spatial pruning ratios. A larger pruning ratio means that fewer voxels remain in the sparse CNN backbone.

Ratio	-	0.1	0.3	0.5	0.7	0.9
FLOPs (G)	83.8	79.6	60.1	33.6	19.8	7.6
mAP	56.5	56.5	56.4	56.2	53.7	45.1
NDS	64.5	64.5	64.3	64.3	62.1	56.0

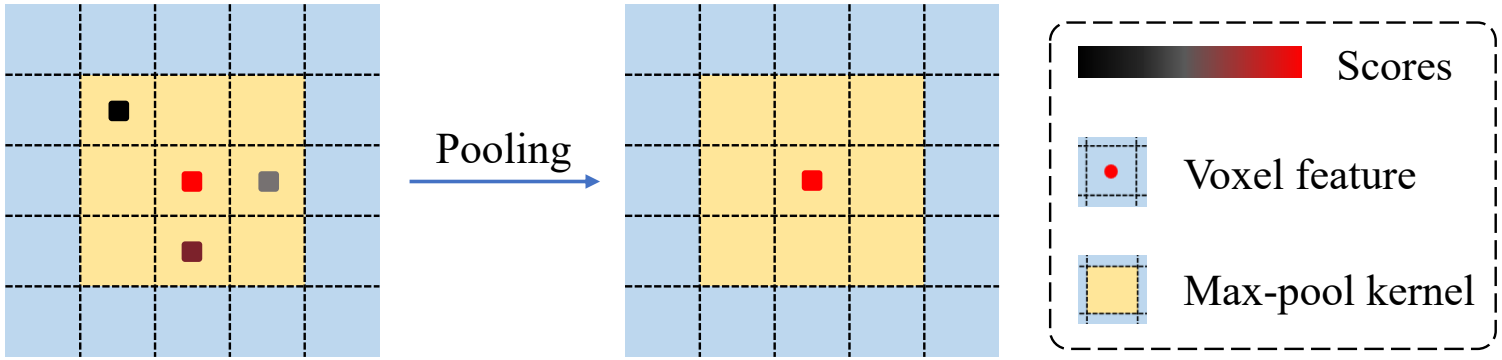
Table 4. Effects of spatial pruning on various layers. We use it on the first 3 down-sampling layers by default.

Stages	-	1	2	3	4	5
FLOPs (G)	83.8	65.0	45.9	33.6	29.1	27.9
mAP	56.5	56.5	56.4	56.2	54.2	53.7
NDS	64.5	64.5	64.4	64.3	62.5	62.0

- **Detail 4: Sparse Max Pooling**

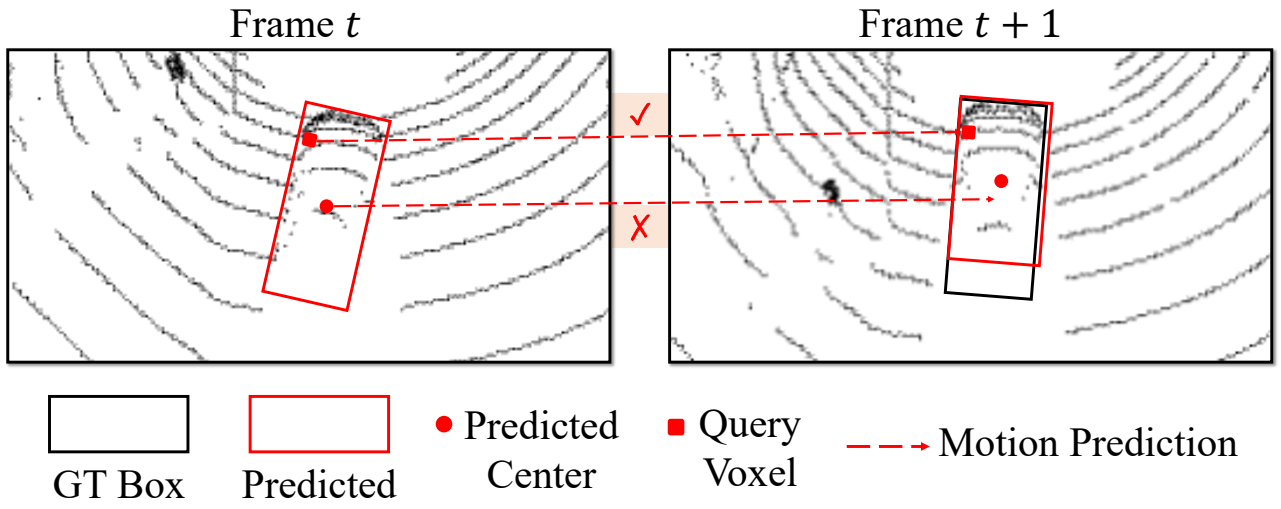


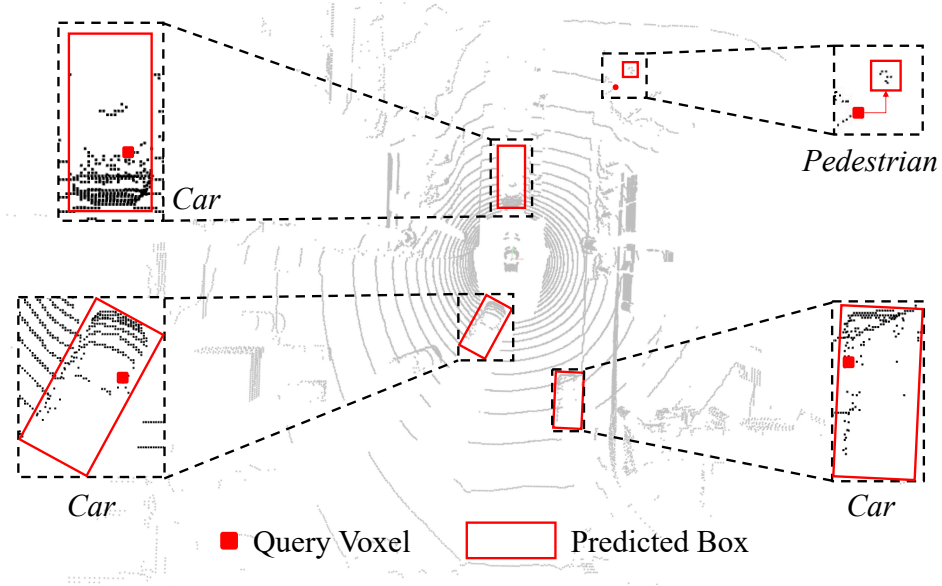
Max-pool	*NMS*	mAP	NDS
✗	✗	33.0	51.0
✗	✓	56.0	64.2
✓	✗	56.2	64.3
✓	✓	56.2	63.3



Sparse max pooling can effectively replace NMS post-processing.

- **Why better tracking?**





- **CenterPoint**: **Center** association for tracking.
- **VoxelNeXt**: **Query voxels** for auxiliary tracking.

➢ Centers are predicted, and might be inaccurate.

➢ *v.s.* Query voxels truly exist in data.

Table 11. Voxel association on nuScenes tracking validation set.

+ *Voxel association*	AMOTA	AMOTP	MOTA	IDS
✗	69.1	61.6	59.3	643
✓	**70.2**	64.0	61.5	729

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Why centers are not required?**

Table 7. Ratios of relative positions of query voxels to the boxes predicted from them. We only take high-quality predicted boxes (IoU with ground-truth boxes > 0.7 and with matched predicted labels) into consideration. According to the relative positions to their predicted boxes, we split voxels into 3 types of *near center*, *near boundary*, and *outside box*. Overall, most voxels are inside but not near center.

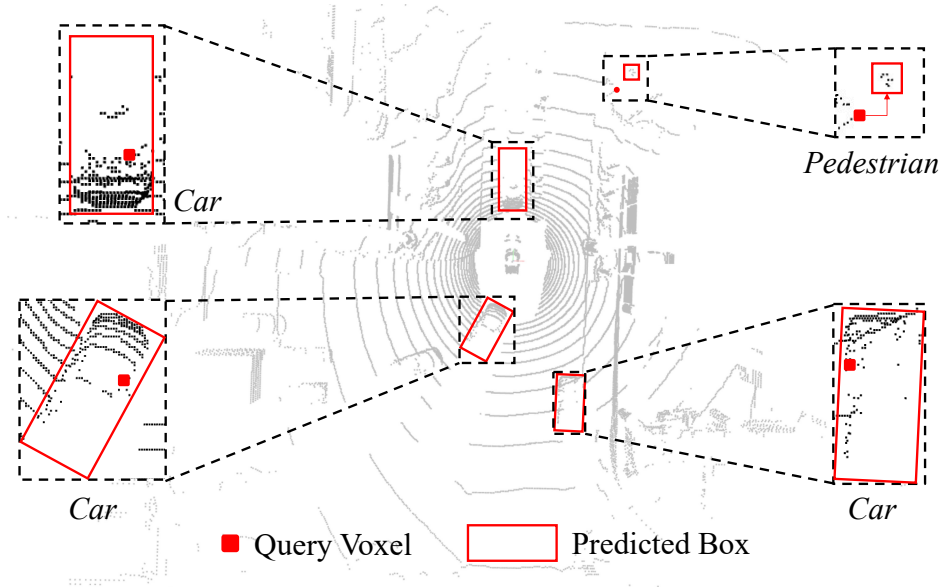
Class	Mean	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
Near center	9.9%	10.3%	5.6%	15.2%	1.2%	16.3%	12.5%	19.6%	13.1%	10.8%	17.8%
Near boundary	**72.8%**	**84.3%**	39.2%	**58.8%**	**84.6%**	**51.8%**	42.3%	**66.5%**	**54.7%**	39.7%	**58.7%**
Outside box	17.3%	5.4%	**55.3%**	26.0%	14.2%	31.9%	**45.2%**	13.9%	32.2%	**49.6%**	23.5%

- *Query voxels are mostly near object boundary.*
- Some are even outside boxes

(Small objects, e.g., Pedestrian, Traffic Cone)

- Few query voxels are near object centers

(LiDAR scatters object surfaces).

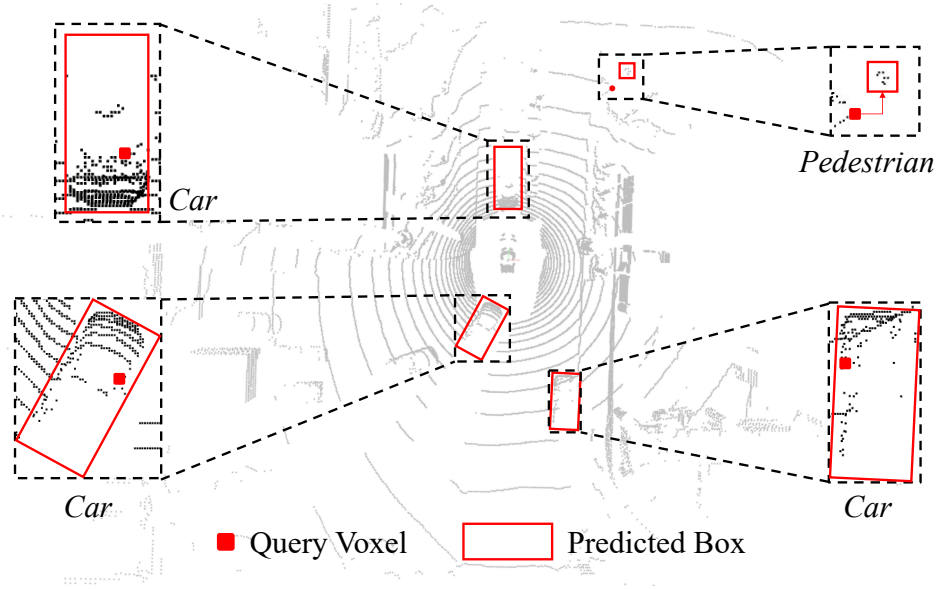


VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Why VoxelNeXt is better than CenterPoint?**

Method	mAP	NDS	ATE	ASE	AOE	AVE	AAE
CenterPoint	55.6	63.5	**29.7**	25.7	44.5	24.5	**18.8**
VoxelNeXt	**56.5**	**64.5**	29.9	**25.4**	**39.6**	**23.2**	19.0
	↑0.9	↑1.0	↑0.2	↓0.3	↓**4.9**	↓1.3	↑0.2

- AOE - Orientation is much better.
- Query voxels is better for angle prediction than centers.
- Relative position of query voxels to centers contain angle information.



VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Results on nuScenes**

Table 12. Performance of 3D object detection methods on nuScenes test set. † means the method that uses double-flip testing. All models listed take LIDAR data as input without image fusion or any model ensemble.

Method	mAP	NDS	Latency	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
PointPillars [20]	30.5	45.3	31 ms	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9
3DSSD [48]	42.6	56.4	-	81.2	47.2	61.4	30.5	12.6	70.2	36.0	8.6	31.1	47.9
CBGS [55]	52.8	63.3	80 ms	81.1	48.5	54.9	42.9	10.5	80.1	51.5	22.3	70.9	65.7
CenterPoint [50]	58.0	65.5	96 ms	84.6	51.0	60.2	53.2	17.5	83.4	53.7	28.7	76.7	70.9
CVCNET [4]	58.2	66.6	122 ms	82.6	49.5	59.4	51.1	16.2	83.0	61.8	38.8	69.7	69.7
HotSpotNet [5]	59.3	66.0	-	83.1	50.9	56.4	53.3	23.0	81.3	63.5	36.6	73.0	71.6
AFDetV2 [19]	62.4	68.5	-	86.3	54.2	62.5	58.9	26.7	85.8	63.8	34.3	80.1	71.0
Focals Conv [7]	63.8	70.0	138 ms	86.7	56.3	67.7	59.5	23.8	87.5	64.5	36.3	81.4	74.1
VISTA [11]†	63.0	69.8	94 ms	84.4	55.1	63.7	54.2	25.1	82.8	70.0	45.4	78.5	71.4
UVTR-L [21]†	63.9	69.7	132 ms	86.3	52.2	62.8	59.7	33.7	84.5	68.8	41.1	74.7	74.9
PillarNet-18 [34]†	65.0	70.8	78 ms	87.4	56.7	60.9	61.8	30.4	87.2	67.4	40.3	82.1	76.0
VoxelNeXt-2D	64.1	69.8	61 ms	84.8	52.7	62.3	56.2	29.5	84.5	72.5	45.7	78.8	73.7
VoxelNeXt	64.5	70.0	66 ms	84.6	53.0	64.7	55.8	28.7	85.8	73.2	45.7	79.0	74.6
VoxelNeXt†	**66.2**	**71.4**	-	85.3	55.7	66.2	57.2	29.8	86.5	75.2	48.8	80.7	76.1

- Good performance on detection.
- SOTA performance on tracking.

Table 13. Performance of nuScenes 3D tracking test split for LIDAR methods. † is based on the double-flip results in Tab. 12.

Method	AMOTA	AMOTP	MOTA	IDS
AB3DMOT [44]	15.1	150.1	15.4	9027
CenterPoint [50]	63.8	55.5	53.7	760
CBMOT [2]	64.9	59.2	54.5	557
OGR3MOT [51]	65.6	62.0	55.4	288
SimpleTrack [30]	66.8	55.0	56.6	575
UVTR-L [21]	67.0	55.0	56.6	774
TransFusion-L [1]	68.6	52.9	57.1	893
VoxelNeXt	**69.5**	56.8	58.6	785
VoxelNeXt†	**71.0**	51.1	60.0	654

Table 14. Performance of nuScenes 3D tracking validation set. All methods listed are LIDAR-only without multi-modal extension.

Method	AMOTA	AMOTP	MOTA	IDS
AB3DMOT [44]	57.8	80.7	51.4	1275
MPN-Baseline	59.3	83.2	51.4	1079
CenterPoint [50]	66.5	56.7	56.2	562
CBMOT [2]	67.5	59.1	58.3	494
OGR3MOT [43]	69.3	62.7	60.2	262
SimpleTrack [30]	69.6	54.7	60.2	405
VoxelNeXt	**70.2**	64.0	61.5	729

- **Results on Waymo**

Method	mAP/mAPH L2	Vehicle		Pedestrian		Cyclist	
		L1 AP/APH	L2 AP/APH	L1 AP/APH	L2 AP/APH	L1 AP/APH	L2 AP/APH
Pillar-OD [50]	-	69.8 / -	- / -	72.5 / -	-	-	-
VoxSeT [21]	-	76.0 / -	68.2 / -	-	-	-	-
VoTr-TSD [34]	-	74.9 / 74.3	65.9 / 65.3	-	-	-	-
SECOND [53]	61.0 / 57.2	72.3 / 71.7	63.9 / 63.3	68.7 / 58.2	60.7 / 51.3	60.6 / 59.3	58.3 / 57.0
M3METR [20]	61.8 / 58.7	75.7 / 75.1	66.0 / 66.0	65.0 / 56.4	56.0 / 48.4	65.4 / 64.2	62.7 / 61.5
IA-SSD [59]	62.3 / 58.1	70.5 / 69.7	61.6 / 61.0	69.4 / 58.5	60.3 / 50.7	67.7 / 65.3	65.0 / 62.7
PointPillars [26]	62.8 / 57.8	72.1 / 71.5	63.6 / 63.1	70.6 / 56.7	62.8 / 50.3	64.4 / 62.3	61.9 / 59.9
RangeDet [17]	65.0 / 63.2	72.9 / 72.3	64.0 / 63.6	75.9 / 71.9	67.6 / 63.9	65.7 / 64.4	63.3 / 62.1
3D-MAN [56]	-	74.5 / 74.0	67.6 / 67.1	71.7 / 67.7	62.6 / 59.0	-	-
LIDAR-RCNN [28]	65.8 / 61.3	76.0 / 75.5	68.3 / 67.9	71.2 / 58.7	63.1 / 51.7	68.6 / 66.9	66.1 / 64.4
PV-RCNN [40]	66.8 / 63.3	77.5 / 76.9	69.0 / 68.4	75.0 / 65.6	66.0 / 57.6	67.8 / 66.4	65.4 / 64.0
Part-A2-Net [43]	66.9 / 63.8	77.1 / 76.5	68.5 / 68.0	75.2 / 66.9	66.2 / 58.6	68.6 / 67.4	66.1 / 64.9
SST [15]	67.8 / 64.6	74.2 / 73.8	65.5 / 65.1	78.7 / 69.6	70.0 / 61.7	70.7 / 69.6	68.0 / 66.9
PV-RCNN++ [41]	68.4 / 64.9	78.8 / 78.2	70.3 / 69.7	76.7 / 67.2	68.5 / 59.7	69.0 / 67.6	66.5 / 65.2
CenterPoint [57]	69.8 / 67.6	76.6 / 76.0	68.9 / 68.4	79.0 / 73.4	71.0 / 65.8	72.1 / 71.0	69.5 / 68.5
AFDetV2 [22]	71.0 / 68.8	77.6 / 77.1	69.7 / 69.2	80.2 / 74.6	72.2 / 67.0	73.7 / 72.7	71.0 / 70.1
PillarNet-34 [39]	71.0 / 68.5	79.1 / 78.6	70.9 / 70.5	80.6 / 74.0	72.3 / 66.2	72.3 / 71.2	69.7 / 68.7
SWFormer [45]	-	77.8 / 77.3	69.2 / 68.8	80.9 / 72.7	72.5 / 64.9	-	-
FSD$_{spconv}$ [16]	71.9 / 69.7	77.8 / 77.3	68.9 / 68.5	81.9 / 76.4	73.2 / 68.0	76.5 / 75.2	73.8 / 72.5
VoxelNeXt-2D	70.9 / 68.2	77.9 / 77.5	69.7 / 69.2	80.2 / 73.5	72.2 / 65.9	73.3 / 72.2	70.7 / 69.6
VoxelNeXt$_{K3}$	**72.2 / 70.1**	78.2 / 77.7	69.9 / 69.4	81.5 / 76.3	73.5 / 68.6	76.1 / 74.9	73.3 / 72.2

VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Results on Argoverse2**

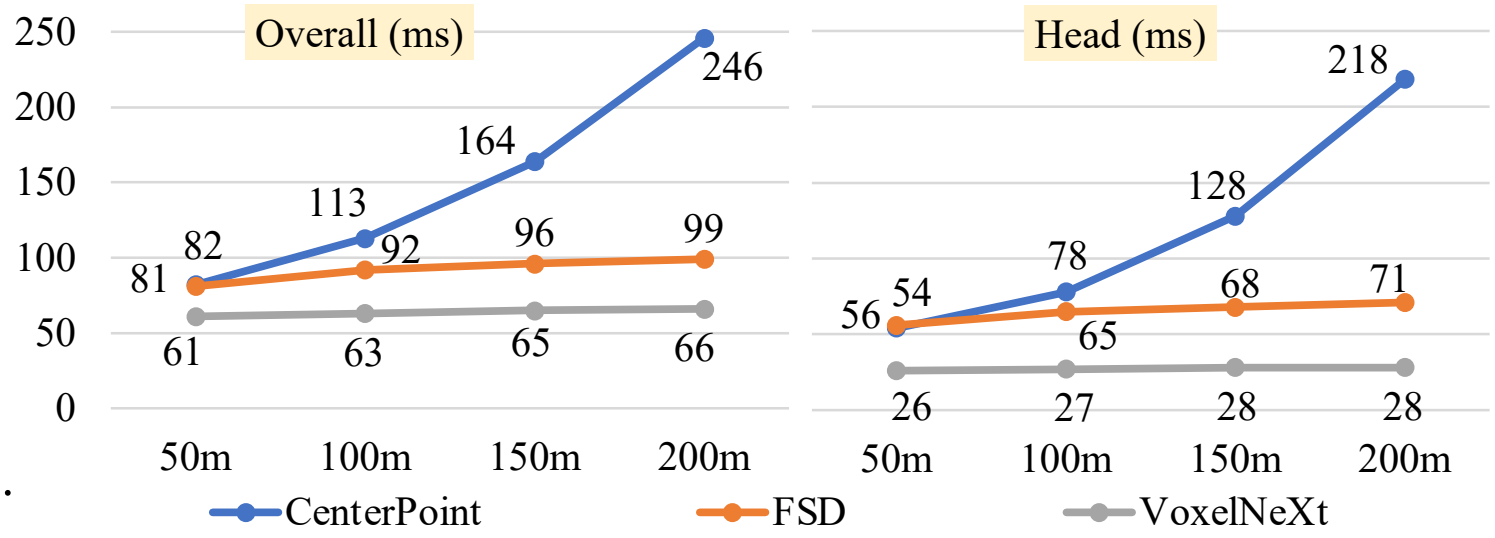
Table 16. Performance of 3D object detection results Argoverse2 dataset.

Methods	mAP	Veh.	Bus	Ped.	Stop.	Box.	Boll.	C-B.	M.-list	MPC.	M.-cycle	Bicycle	A-B.	School.	Truck.	C-C.	V-T.	Sign	Large.	Str.	Bic.-list
CenterPoint [50]	22.0	67.6	38.9	46.5	16.9	37.4	40.1	32.2	28.6	27.4	33.4	24.5	8.7	25.8	**22.6**	29.5	22.4	6.3	3.9	0.5	20.1
FSD	28.2	68.1	**40.9**	59.0	29.0	38.5	41.8	42.6	39.7	26.2	**49.0**	38.6	**20.4**	**30.5**	14.8	41.2	**26.9**	11.9	5.9	13.8	**33.4**
VoxelNeXt	**30.7**	**72.7**	38.8	**63.2**	**40.2**	**40.1**	**53.9**	**64.9**	**44.7**	**39.4**	42.4	**40.6**	20.1	25.2	19.9	**44.9**	20.9	**14.9**	**6.8**	**15.7**	32.4

- Argoverse2: 400m x 400m.

v.s. KITTI, nuScenes, Waymo: 75m radius.

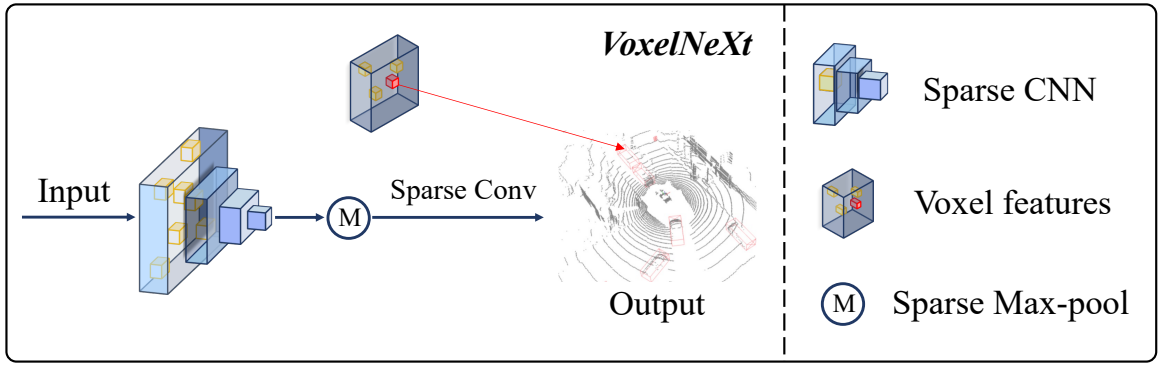
- SOTA performance and efficiency.
- Better than the voting-based FSD [1].
- Consistent efficiency to perception range.

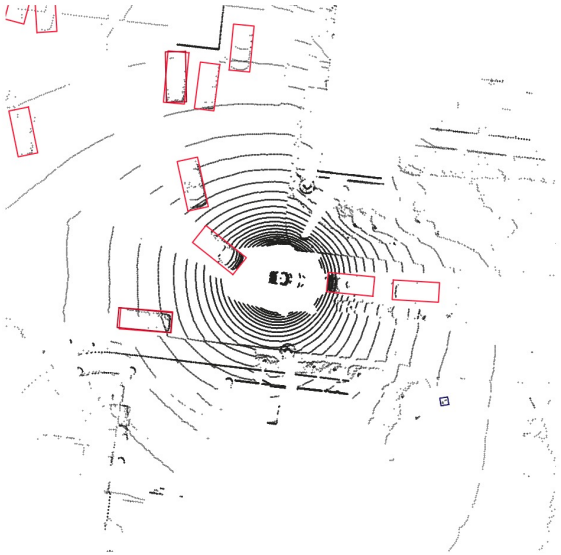


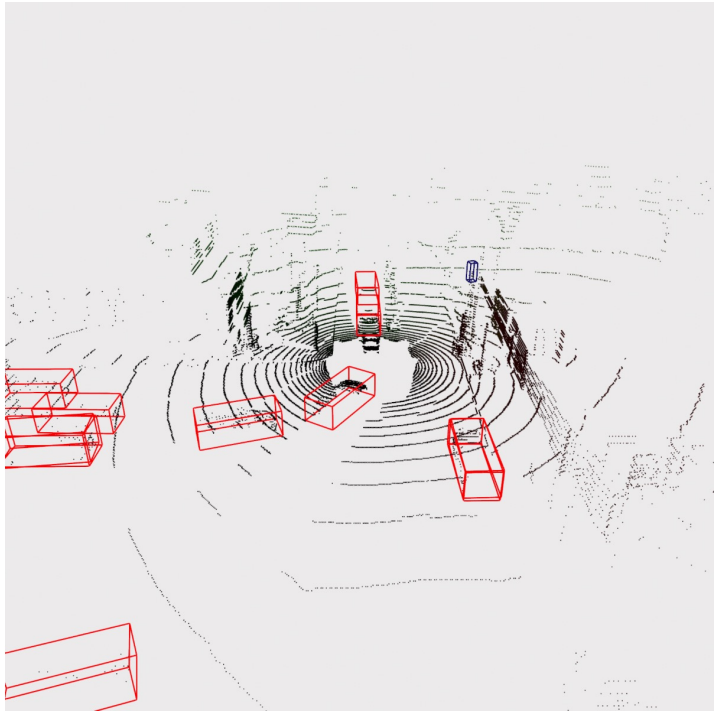
Lue Fan, Feng Wang, Naiyan Wang, Zhaoxiang Zhang, Fully Sparse 3D Object Detection, NeurIPS 2022

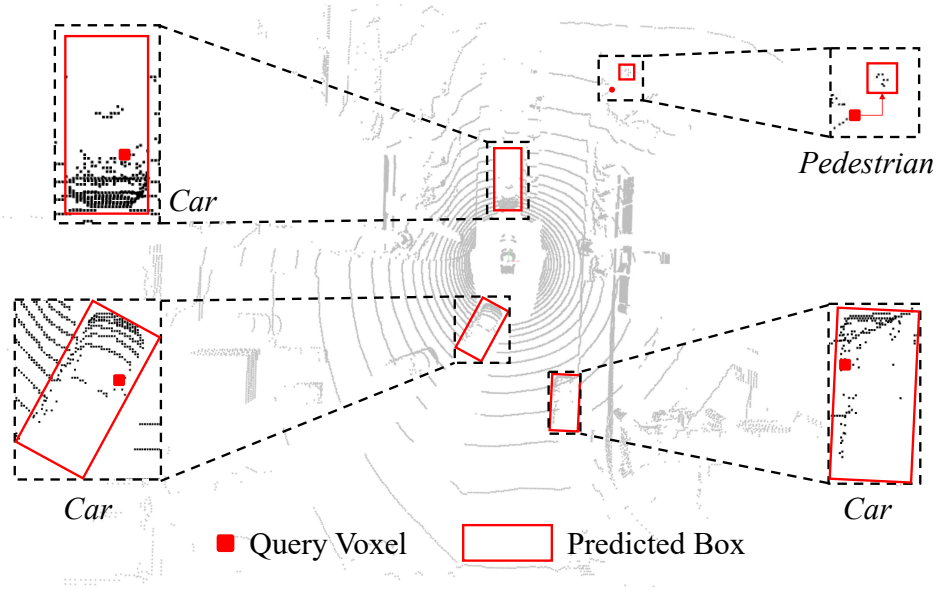
VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking (CVPR 2023)

- **Conclusion**
  - A very simple 3D detection pipeline
  - **High performance & efficiency**
  - **nuScenes & Waymo & Argoverse2**



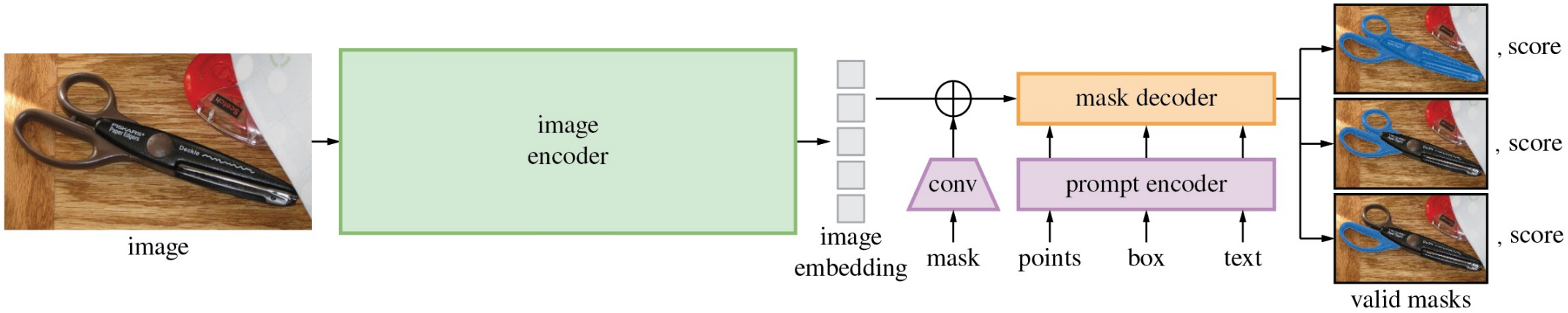






Segment Anything + VoxelNeXt

- **Segment Anything**



Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alex Berg, Wan-Yen Lo, Piotr Dollar, Ross Girshick. Segment Anything

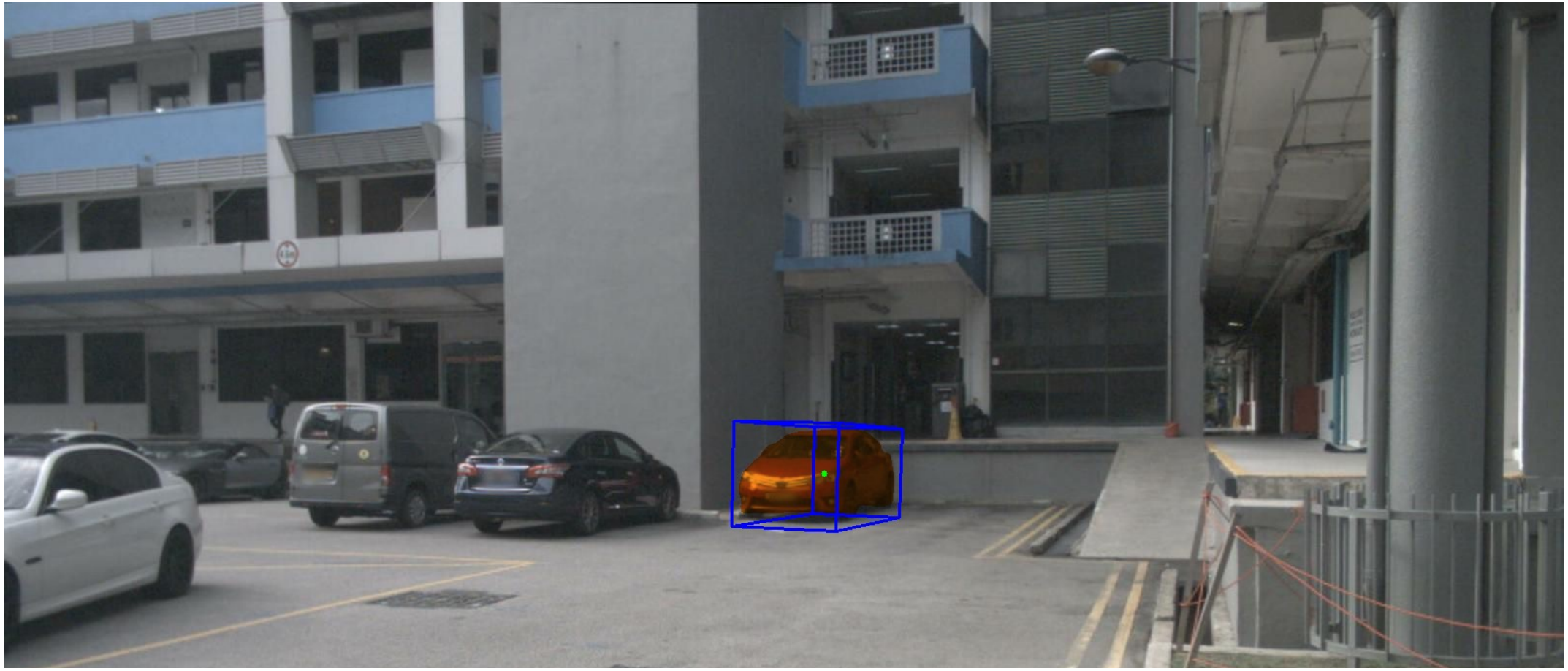
- **Segment Anything**

Segment Anything & VoxelNeXt

- **Sparse voxel in a mask → 3D Object**

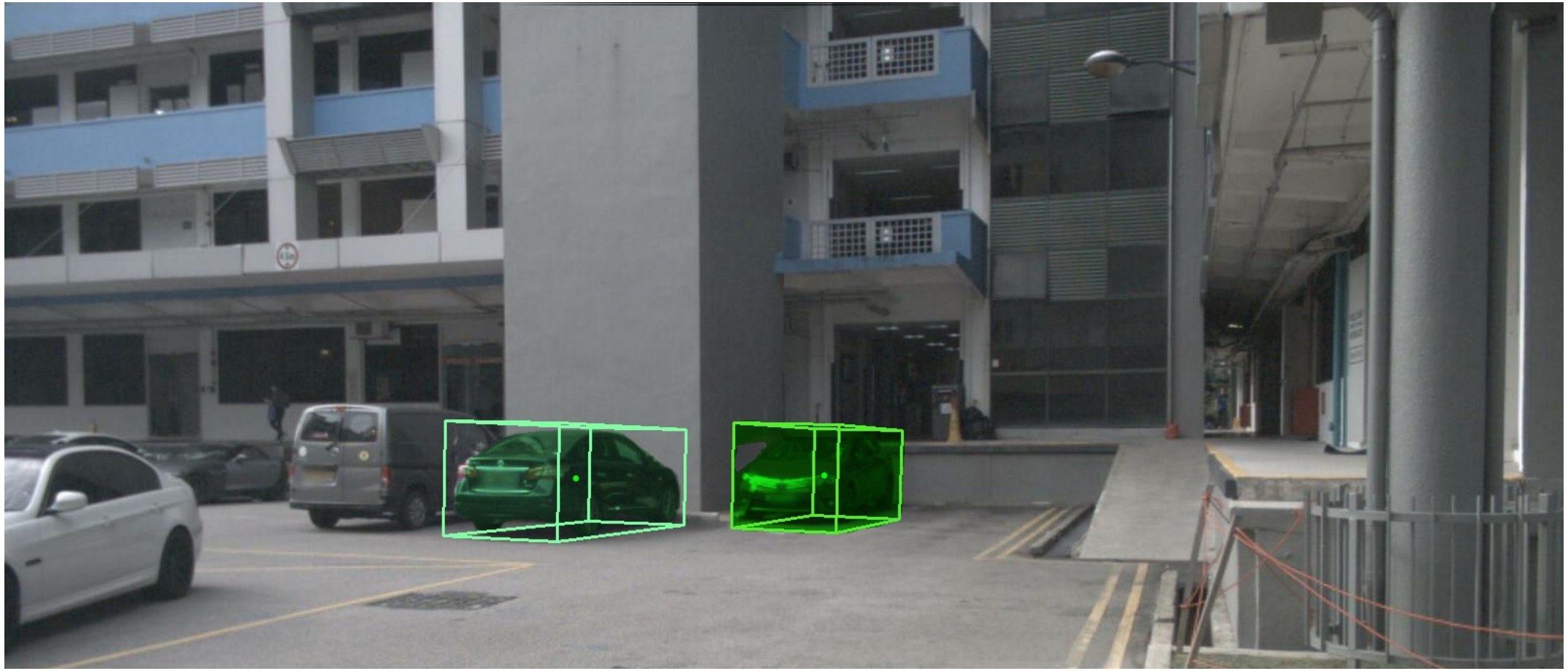
Segment Anything & VoxelNeXt

- **Sparse voxel in a mask → 3D Object**



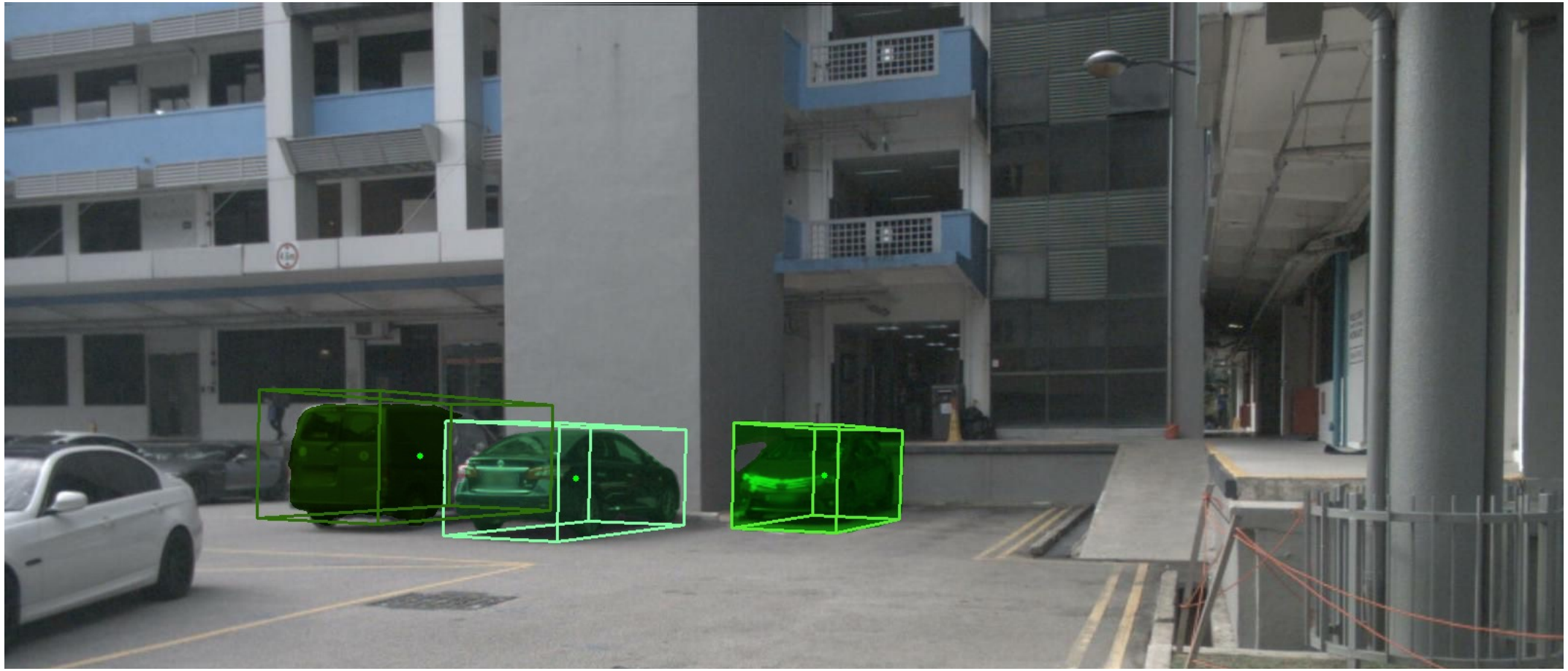
Segment Anything & VoxelNeXt

- **Sparse voxel in a mask → 3D Object**



Segment Anything & VoxelNeXt

- **Sparse voxel in a mask → 3D Object**



Reference

[1] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In CVPR, pages 9224–9232, 2018.

[2] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: pointvoxel feature set abstraction for 3d object detection. In CVPR, pages 10526–10535, 2020.

[3] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: towards high performance voxel-based 3d object detection. In AAAI, pages 1201–1209, 2021.

[4] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Centerbased 3d object detection and tracking. In CVPR, pages 11784–11793, 2021.

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. Int. J. Robotics Res., 32(11):1231–1237, 2013.

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In CVPR, pages 11618–11628, 2020.

Yukang Chen

- Third-year Ph.D student in CUHK
- Supervised by Jiaya Jia
- Research in Efficient Computer Vision
  - *AutoML, Autonomous driving, Multi-modality*
- More about me
  - https://yukangchen.com
  - https://scholar.google.com/citations?user=6p0ygKUAAAAJ&hl=en

- Thanks!