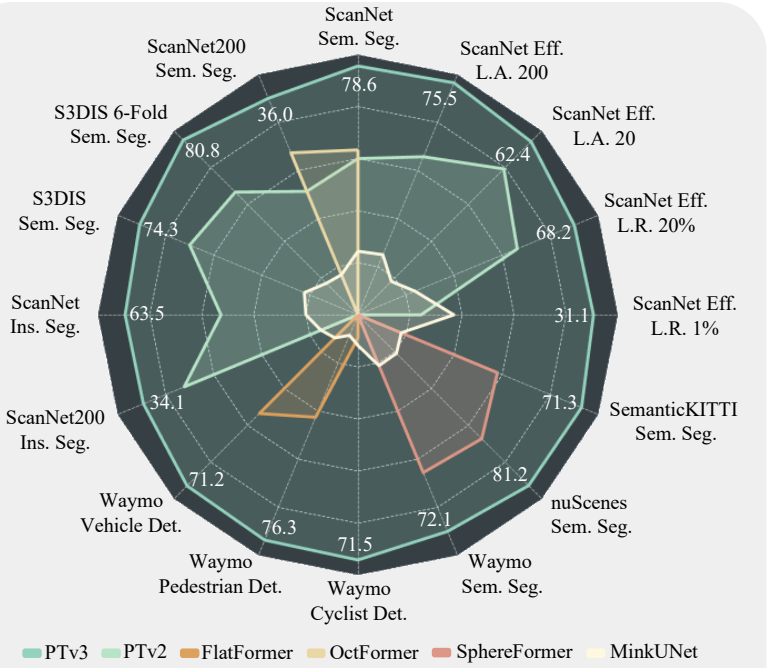
Point Transformer V3: Simpler, Faster, Stronger

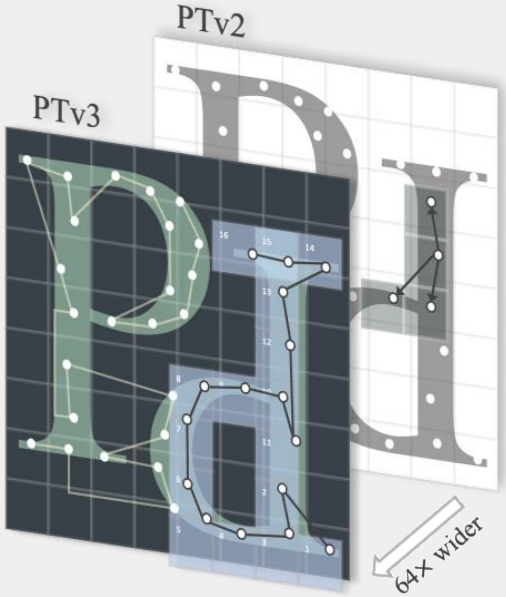
⇒ State-of-the-art performance on over **20** downstream tasks that span both **indoor** and **outdoor** scenarios.

⇒ Expanding the receptive field from **16** to **1024** points while remaining efficient.

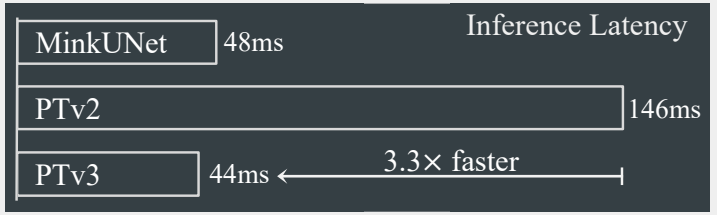
⇒ **3x** increase in processing speed and **10x** improvement in memory efficiency compared with PTv2



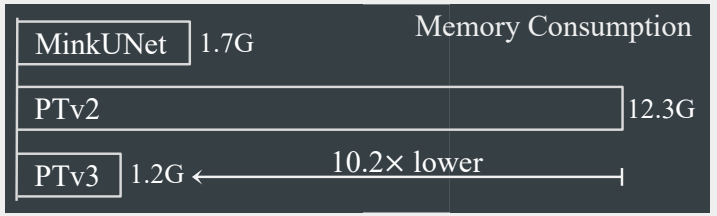
Stronger Performance



Wider Receptive Field



Faster Speed



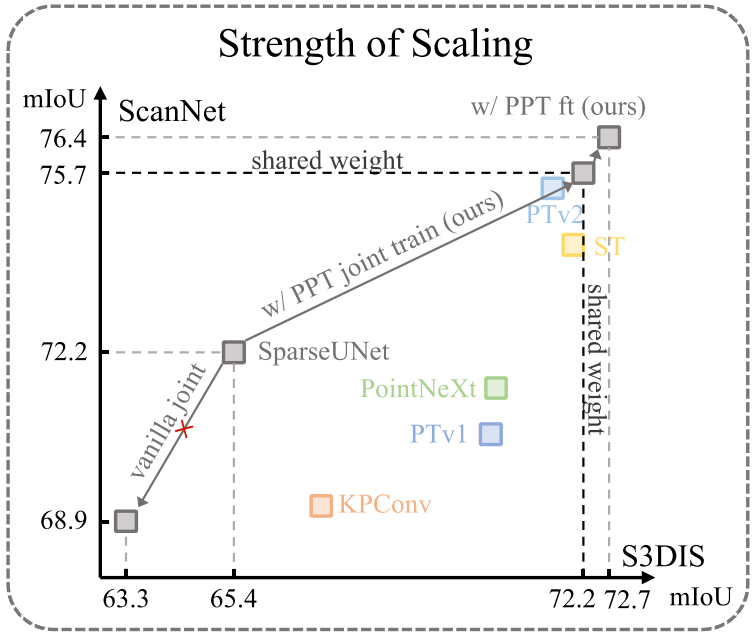
Lower Memory Consumption

1st Place Solution for 2024 **Waymo Open Dataset** Challenge in Semantic Segmentation

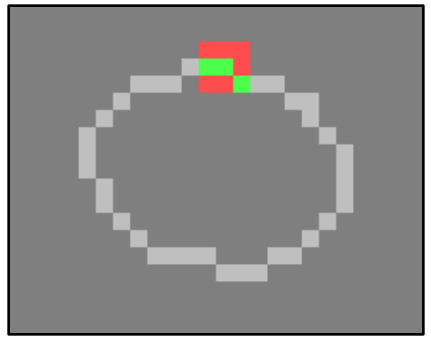
POINTCEPT
Point Cloud Perception Codebase

香 港 大 學
THE UNIVERSITY OF HONG KONG

Point Transformer V3: Scaling Principle

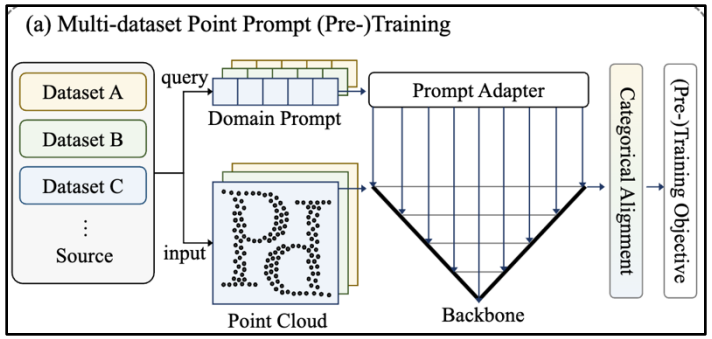


From *Towards Large-scale 3D Representation Learning with Multi-dataset Point Prompt Training*



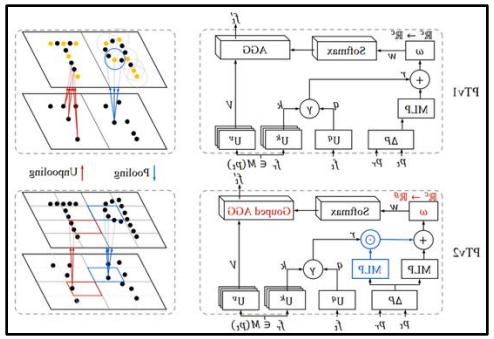
SparseUNet (2019)

+



PPT (Large-scale Training)

>



PTv2 (2022)

⇒ Enhanced with large-scale pre-training, **SparseUNet** surpasses **Point Transformers** in accuracy while remaining efficient;

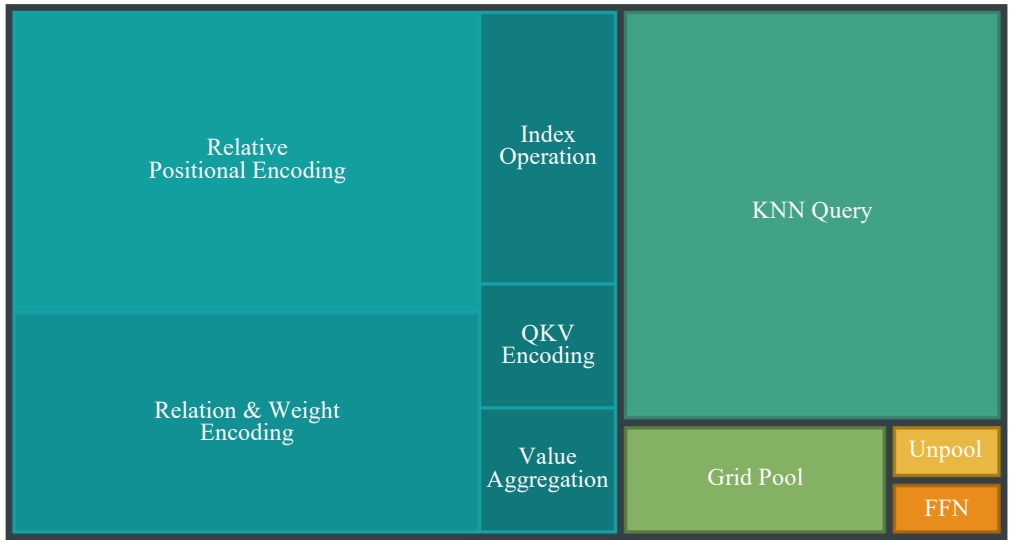
⇒ **Point Transformers** fails to scale up due to limitations in efficiency;

⇒ We hypothesize that model performance is **more significantly influenced by scale** than by complex design details;

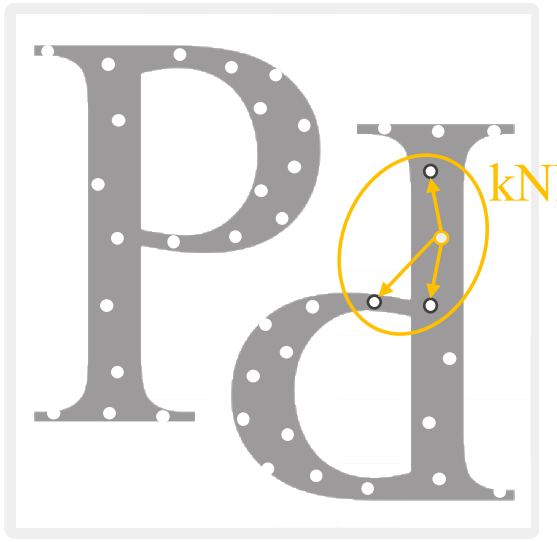
⇒ We should **prioritize simplicity and efficiency over the accuracy** of certain mechanisms;

⇒ **Efficiency enable scalability** and further enable stronger accuracy.

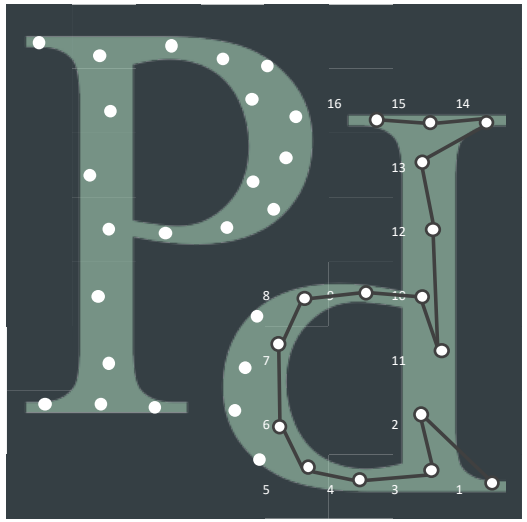
Point Transformer V3: breaking the curse of permutation invariance



PTv2 Forward Latency



PTv2



PTv3

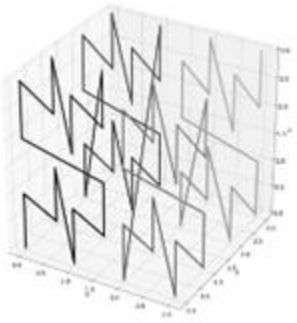
⇒ Classical point cloud transformers build upon point-based backbones, which treat point clouds as **unstructured data** and rely on neighboring query algorithms like **kNN**;

⇒ Yet kNN is extremely inefficient due to difficulty in parallelization. (28% latency)

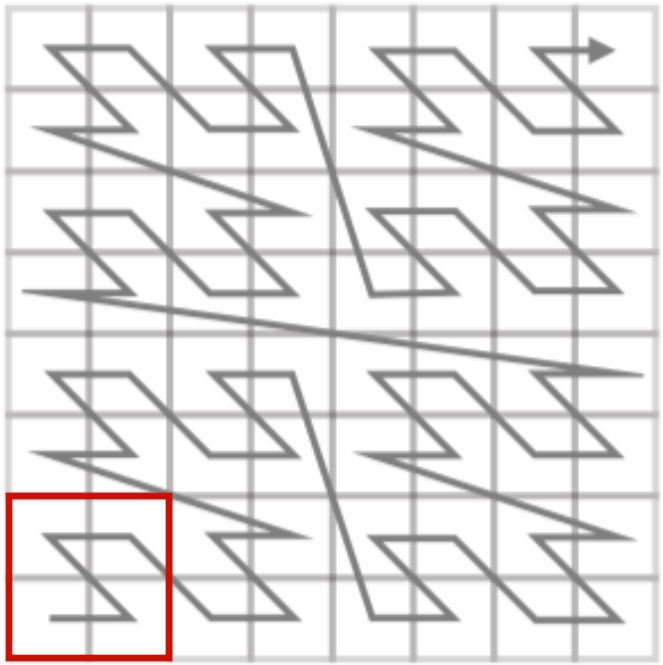
⇒ **Do we really need the accurate neighbors** queried by kNN? **No**, attention is adaptive to kernel shape, all we need to do is relatively precise and **enlarge the kernel shape**;

⇒ Inspired by OctFormer and FlatFormer, we move away from the traditional paradigm, which treats point clouds as unordered sets. we choose to **break the curse by serializing point cloud into a structured format**.

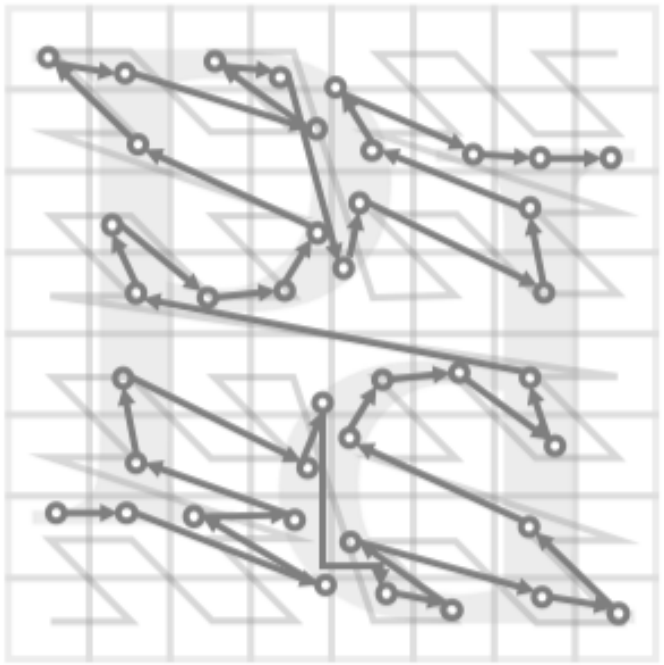
Point Transformer V3: making unstructured sparse data structured!



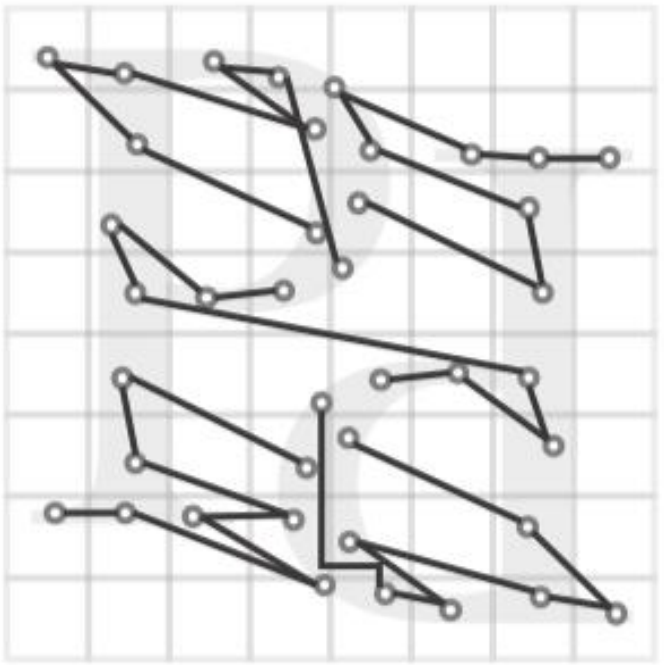
Do precise neighborhood by KNN really matter?
=> No! Attention is adaptive! We just need to make sure the kernel is large! 16 => 1024



Space-filling Curve



Ordered Point Cloud
structured 1D format

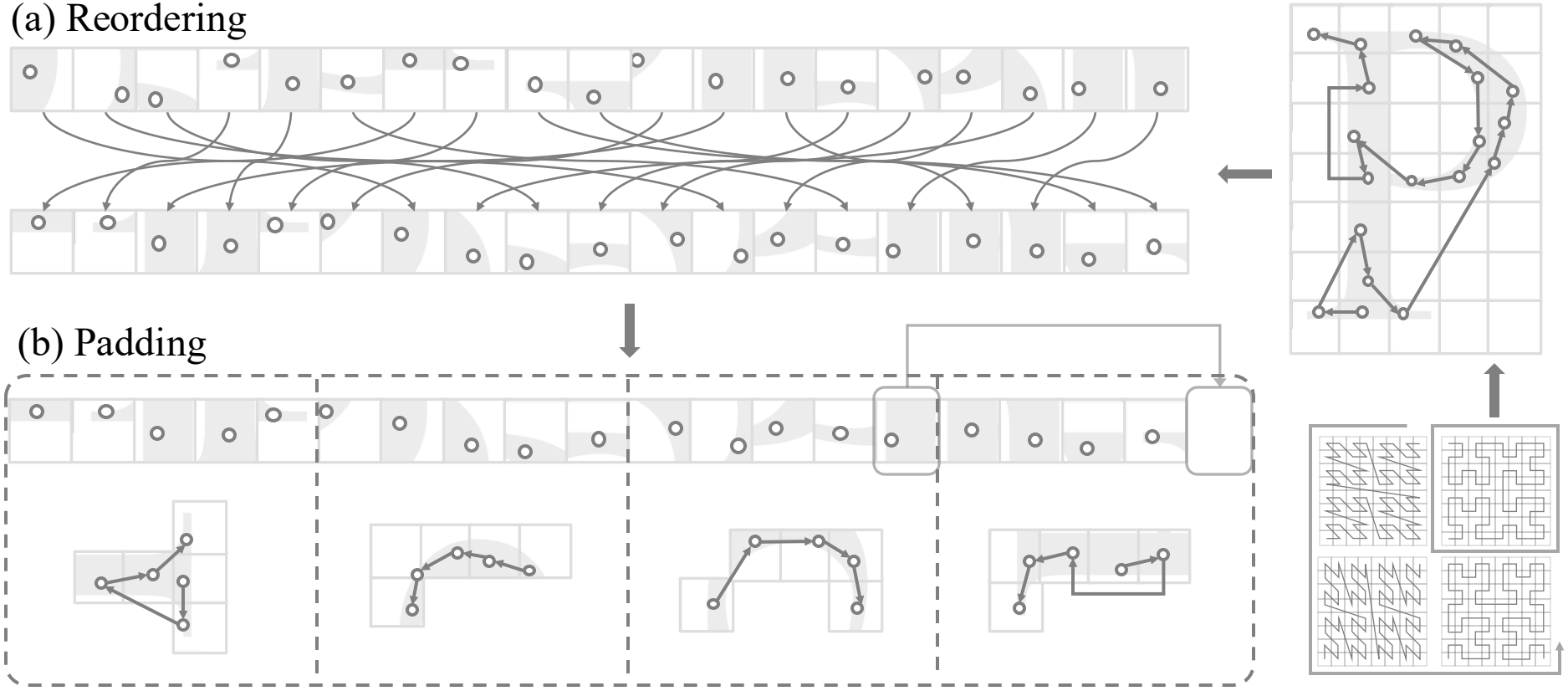


Patch Partition
preserving spatial
neighbor relationships

Method

Point Transformer V3: Serialized Attention

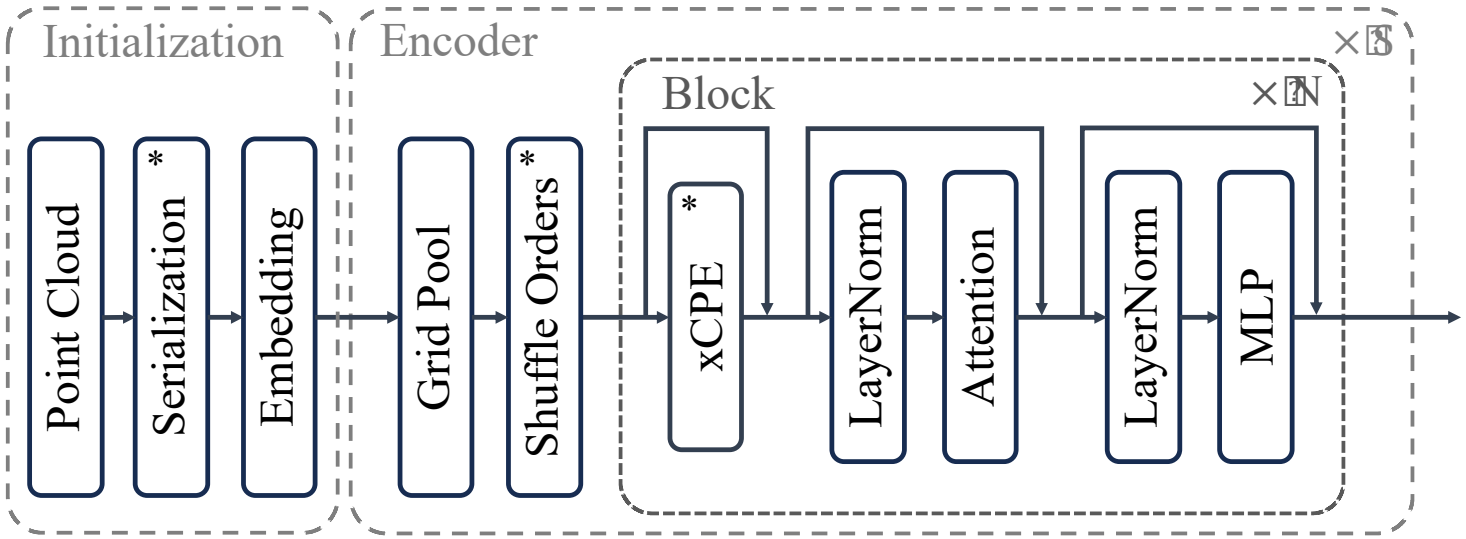
(a) Reordering

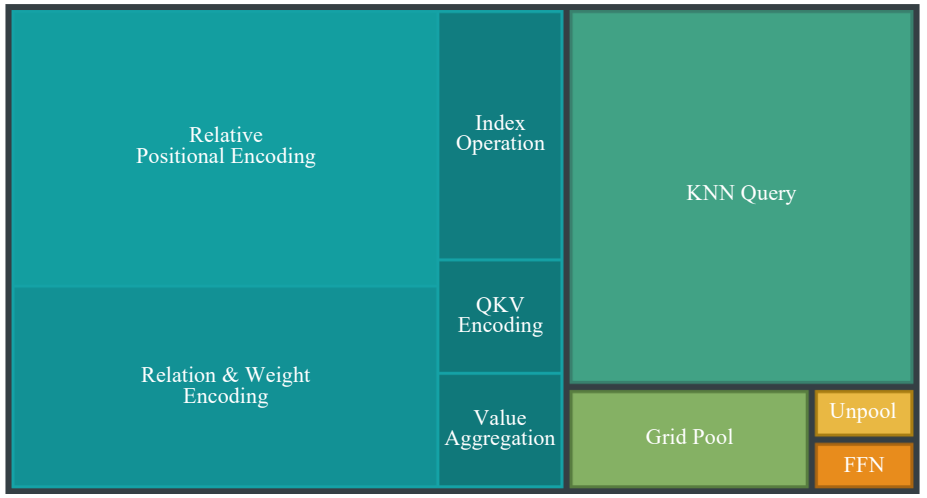


(b) Padding

⇒ **Computing the order** of given unstructured point cloud and space-filling curve pattern.
⇒ **Making** the point cloud **structured** with the computed order after padding.
⇒ Then the unstructured data is arranged as an 1D array just **as language tokens**.
⇒ We can directly **apply** well-optimized **attention operator** designed for structured data.

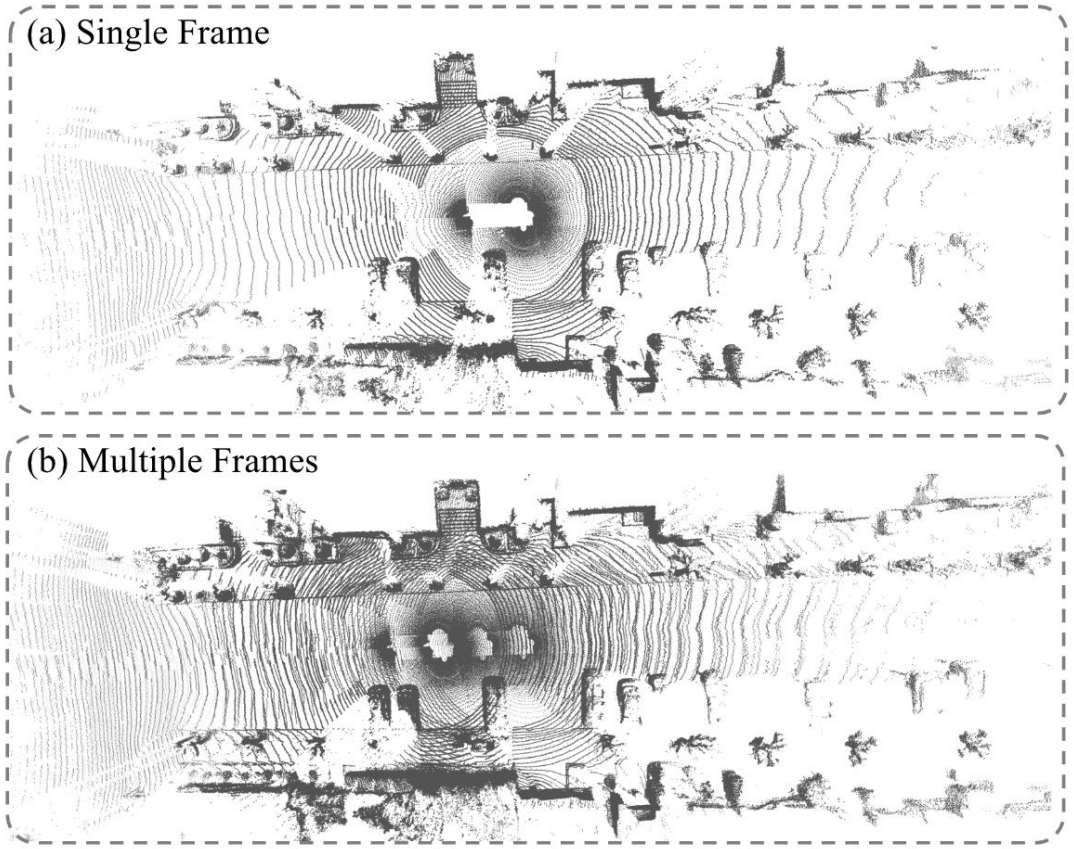
Point Transformer V3: xCPE





⇒ **Relative positional encoding** is also time-consuming for point cloud transformers.
⇒ Use **Sparse Convolution** as a replacement for relative positional encoding (xCPE).
⇒ If Sparse Convolution is not easy to deploy or CPU only inference,
⇒ Use **O-CNN PyTorch** by Peng Shuang Wang as a replacement (Full PyTorch Code)
⇒ https://github.com/octree-nn/ocnn-pytorch

Point Transformer V3 - Extreme



⇒ **Multi-frame Training.** Incorporate two past labelled frames as additional references during both our training and inference processes.

⇒ **Non-clipping proxy.** Clipping points to a specific range was a necessary preprocessing step for perception tasks in outdoor scenarios limited by Sparse Convolution

⇒ **Model ensemble.** Independently train three PTv3 models and combine their predicted logits to form our final submission.

Point Transformer V3 - Performance

Original		Extreme	
Config	Value	Config	Value
optimizer	AdamW	optimizer	AdamW
scheduler	Cosine	scheduler	Cosine
criteria	CrossEntropy (1) Lovasz [1] (1)	criteria	CrossEntropy (1) Lovasz [1] (1)
learning rate	2e-3	learning rate	2e-3
block lr scaler	1e-1	block lr scaler	1e-1
weight decay	5e-3	weight decay	5e-3
batch size	12	batch size	12
datasets	Waymo	datasets	Waymo
warmup epochs	2	warmup epochs	2
epochs	50	epochs	50
frames	[0]	frames	[0, -1, -2]
model ensemble	×	model ensemble	✓

Table 1. **Training settings.**

Augmentations	Parameters	Original	Extreme
random rotate	axis: z, angle: [-1, 1], p: 0.5	✓	✓
point clip	range: [-75.2, -75.2, -4, 75.2, 75.2, 2]	✓	×
random scale	scale: [0.9, 1.1]	✓	✓
random flip	p: 0.5	✓	✓
random jitter	sigma: 0.005, clip: 0.02	✓	✓
grid sampling	grid size: 0.05	✓	✓

Table 2. **Data augmentations.**

Sem. Seg.	PTv3 [15]		PTv3-EX	
	val	test	val	test
Model Ensemble	-	✓	-	✓
Params.	46.2M	46.2M×3	46.2M	46.2M×3
Training Latency	245ms	245ms×3	482ms	482ms×3
Inference Latency	132ms	132ms×3	253ms	253ms×3
Car	0.9447	0.9571	0.9463	0.9662
Truck	0.6207	0.6793	0.6283	0.7397
Bus	0.8665	0.7482	0.8920	0.7792
Other Vehicle	0.3582	0.3654	0.4857	0.3681
Motorcyclist	0.1630	0.0000	0.3946	0.1514
Bicyclist	0.7878	0.9010	0.8030	0.9203
Pedestrian	0.9120	0.9264	0.9162	0.9372
Sign	0.7235	0.7404	0.7664	0.7502
Traffic Light	0.3607	0.3373	0.4276	0.3465
Pole	0.7778	0.8157	0.8036	0.8254
Construction Cone	0.7562	0.6690	0.7405	0.6693
Bicycle	0.7821	0.6851	0.7772	0.7226
Motorcycle	0.9034	0.8070	0.9154	0.8263
Building	0.9606	0.9736	0.9636	0.9751
Vegetation	0.9189	0.8812	0.9242	0.8901
Tree Trunk	0.6860	0.7500	0.7069	0.7575
Curb	0.7152	0.7520	0.7226	0.7648
Road	0.9348	0.9306	0.9368	0.9330
Lane Marker	0.5712	0.4967	0.5726	0.5111
Other Ground	0.5206	0.5255	0.5248	0.5414
Walkable	0.8167	0.7357	0.8196	0.7538
Sidewalk	0.7872	0.8733	0.7891	0.8788
mIoU	0.7213	0.7068	**0.7480**	**0.7276**

Point Transformer V3 - Information

Oral

Point Transformer V3: Simpler Faster Stronger

Xiaoyang Wu · Li Jiang · Peng-Shuai Wang · Zhijian Liu · Xihui Liu · Yu Qiao · Wanli Ouyang · Tong He · Hengshuang Zhao

Summit Flex Hall C Oral #1

[Abstract] [Visit Orals 2C 3D from multiview and sensors]
Wed 19 Jun 1 p.m. — 1:18 p.m. PDT (Bookmark)

Poster presentation: Point Transformer V3: Simpler Faster Stronger
Wed 19 Jun 5 p.m. PDT — 6:30 p.m. PDT (Bookmark)

[OpenReview]

[Paper Metadata for Authors (e.g. Slide Uploads...)]

Tutorial

All You Need To Know About Point Cloud Understanding

Xiaoyang Wu · Hengshuang Zhao · Fuxin Li · Zhijian Liu

Summit 444

[Abstract] [Project Page] |
Tue 18 Jun 9 a.m. PDT — 12:15 p.m. PDT (Bookmark)