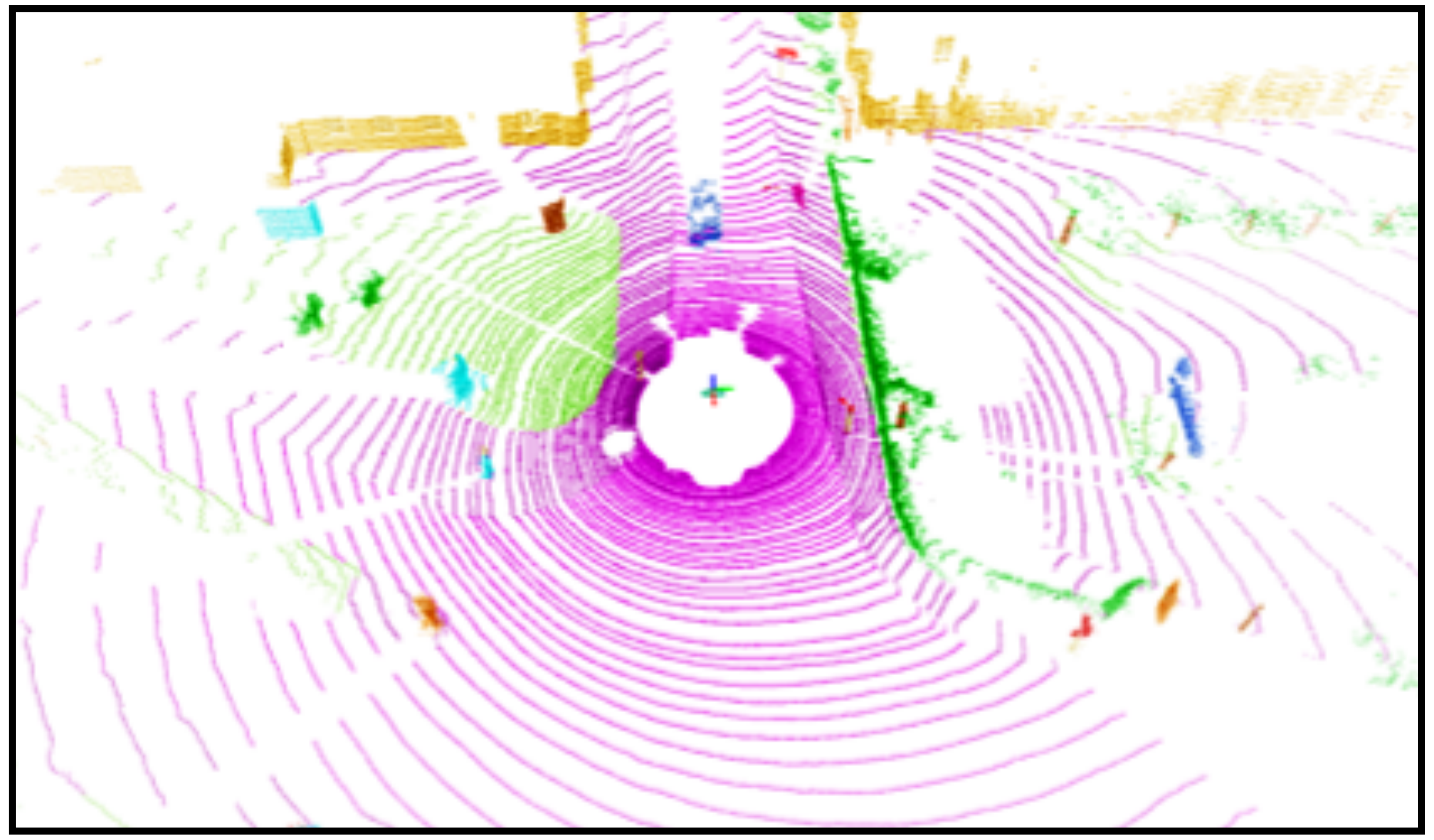
FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer

Zhijian Liu[1,#], Xinyu Yang[2,#], Haotian Tang[1], Shang Yang[3], Song Han[1]

[1]: MIT [2]: Shanghai Jiao Tong University [3]: Tsinghua University

Background

Point cloud deep learning is widely used in real-world applications



Data: 3D Point Cloud

Autonomous Driving

Augmented Reality / Mixed Reality

Applications

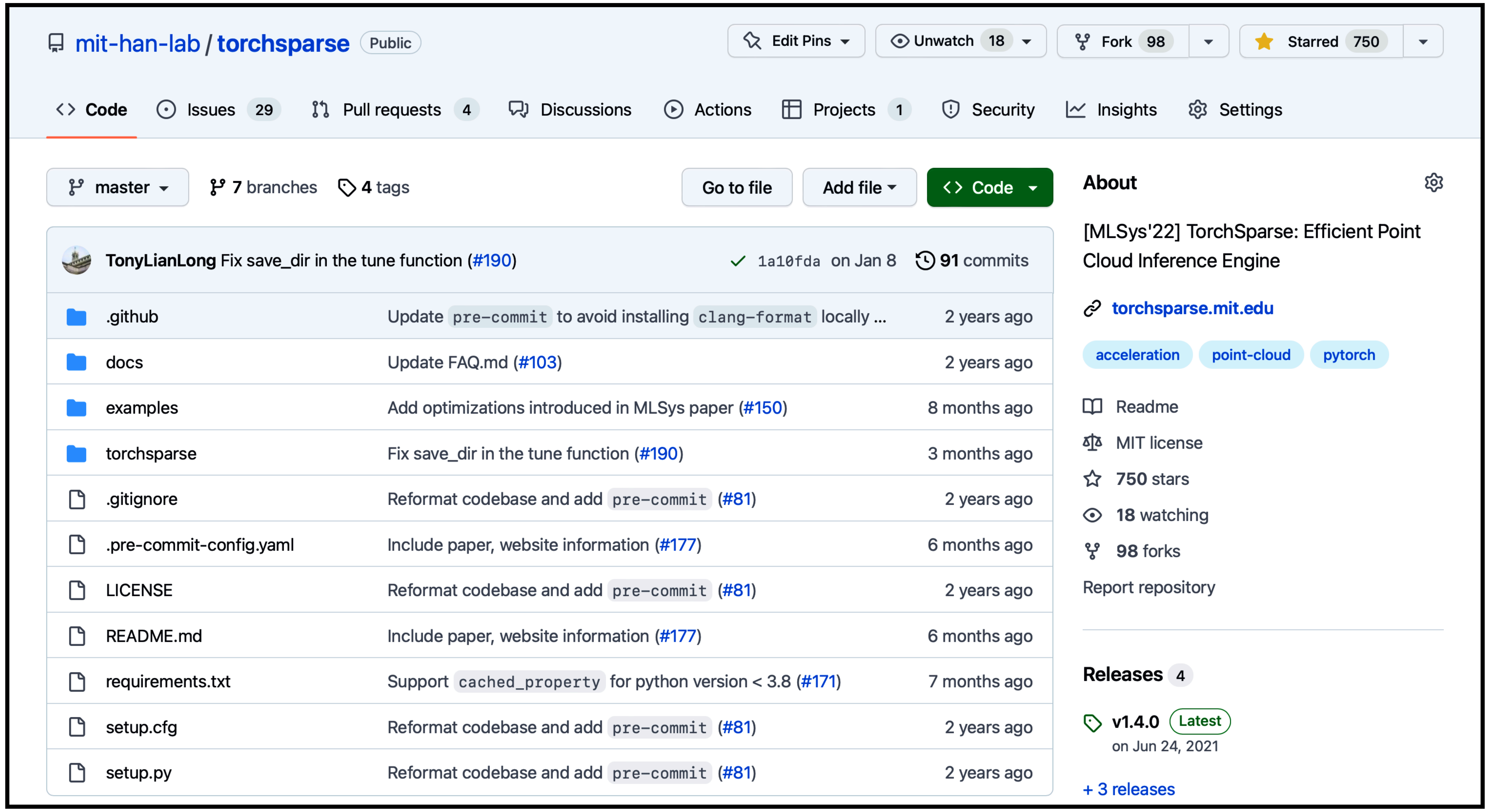
Background

State-of-the-art point cloud models are based on sparse convolution

	Method					Metrics									
	Date	Name	Modalities	Map data	External data	mAP	mATE (m)	mASE (1-IOU)	mAOE (rad)	mAVE (m/s)	mAAE (1-acc)	NDS	PKL *	FPS (Hz)	Stats
			Lidar	*All*	*All*										
›	2023-03-10	FocalFormer3D-TTA	Lidar	no	no	0.705	0.243	0.238	0.321	0.200	0.131	0.739	0.564	n/a	
›	2023-03-06	Real-Aug	Lidar	no	no	0.689	0.237	0.227	0.320	0.184	0.131	0.734	0.597	n/a	
›	2022-10-31	LinK	Lidar	no	no	0.698	0.238	0.229	0.312	0.234	0.136	0.734	0.575	n/a	
›	2022-05-30	LargeKernel-L	Lidar	no	no	0.688	0.244	0.230	0.312	0.241	0.132	0.728	0.581	n/a	
›	2022-11-11	SphereFormer	Lidar	no	no	0.685	0.242	0.233	0.353	0.191	0.131	0.728	0.616	n/a	
›	2022-09-27	MDRNet-L	Lidar	no	no	0.684	0.242	0.229	0.319	0.227	0.129	0.728	0.581	n/a	
›	2021-07-30	DAA_AVP	Lidar	no	no	0.697	0.237	0.229	0.376	0.248	0.124	0.727	0.937	n/a	
›	2022-08-18	MGTANet	Lidar	no	no	0.675	0.250	0.233	0.308	0.193	0.124	0.727	0.538	n/a	
›	2023-03-10	FocalFormer3D	Lidar	no	no	0.687	0.254	0.242	0.340	0.218	0.126	0.726	0.586	n/a	
›	2021-04-19	CenterPoint-VID	Lidar	no	no	0.674	0.255	0.235	0.339	0.233	0.128	0.718	0.555	n/a	

Background

Sparse convolution requires specialized system support to run on GPUs




```python
from torch import nn
from torchsparse import nn as tsnn
from torchsparse import SparseTensor

# Model definition
model = nn.Sequential(
  tsnn.Conv3d(4, 16, 3, stride=2, padding=1),
  tsnn.BatchNorm(16),
  tsnn.ReLU(True),
).cuda()

# Tensor definition
x = SparseTensor(feats, coords).cuda()

# Forward and backward pass
y = model(x)
loss = criterion(y, ...)
loss.backward()
```

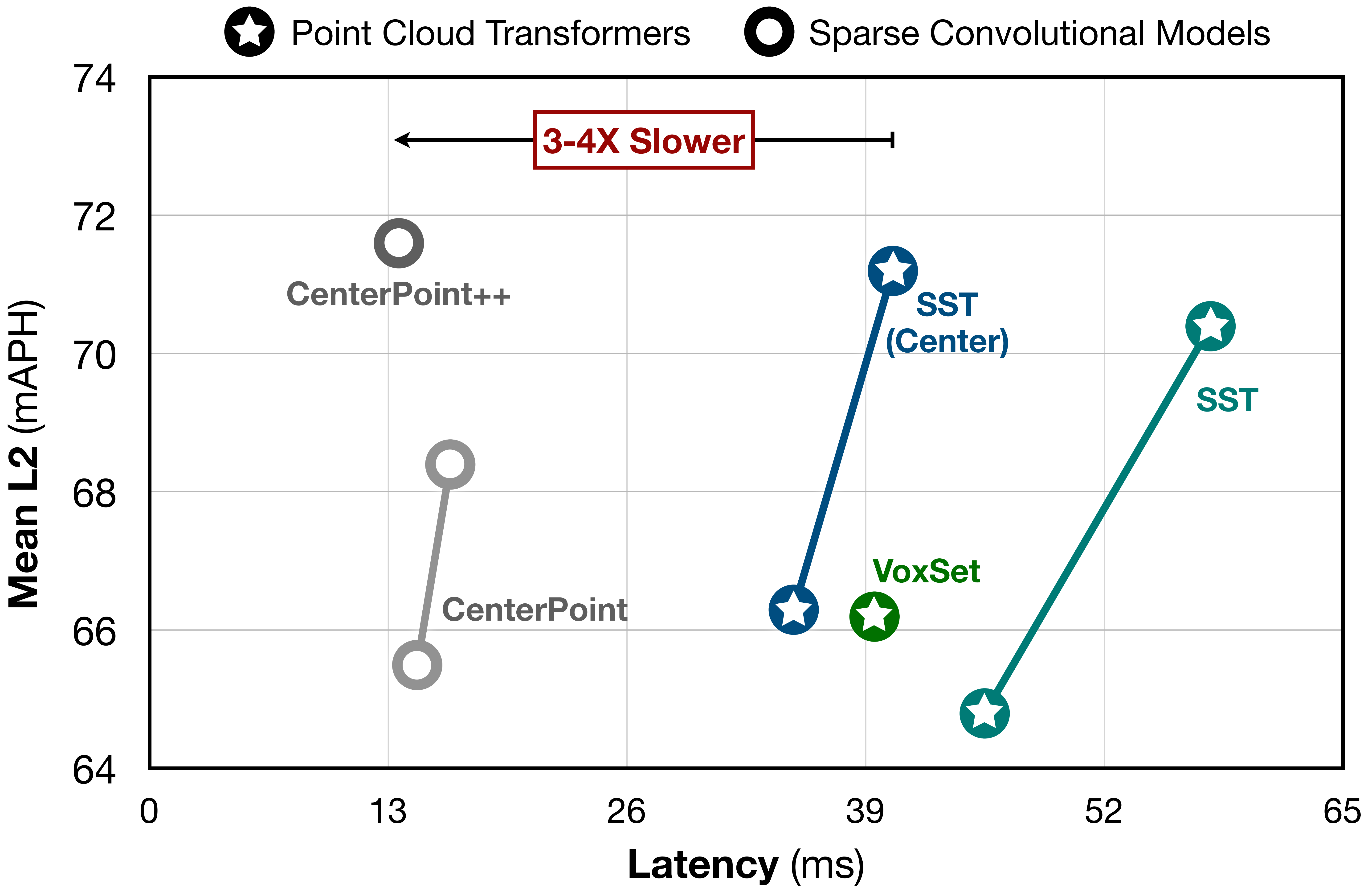
Installation:

```
pip install --upgrade git+https://github.com/mit-han-lab/torchsparse.git
```

TorchSparse: Efficient Point Cloud Inference Engine [Tang et al., MLSys 2022]

Point Cloud Transformers

Achieve comparable accuracy but lag far in latency (3-4X slower)



Point Cloud Transformers

Achieve comparable accuracy but lag far in latency (3-4X slower)

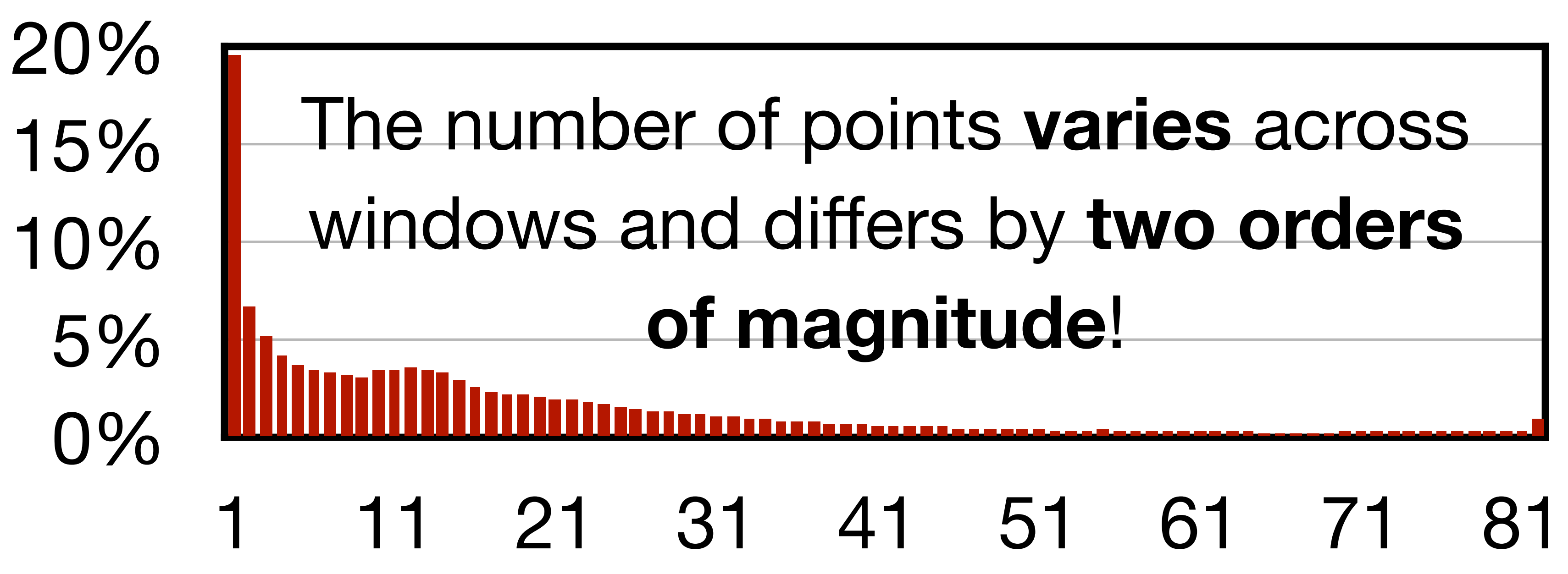
Global PCTs: Apply MHSA globally across the entire point cloud.

The runtime of global PCTs grows **quadratically** as the number of points grows.

The model takes almost **1 second** to run with **32k** input points.

Guo et al., "PCT: Point Cloud Transformer", CVM 2021

Window PCTs: Apply MHSA to a set of non-overlapping windows.



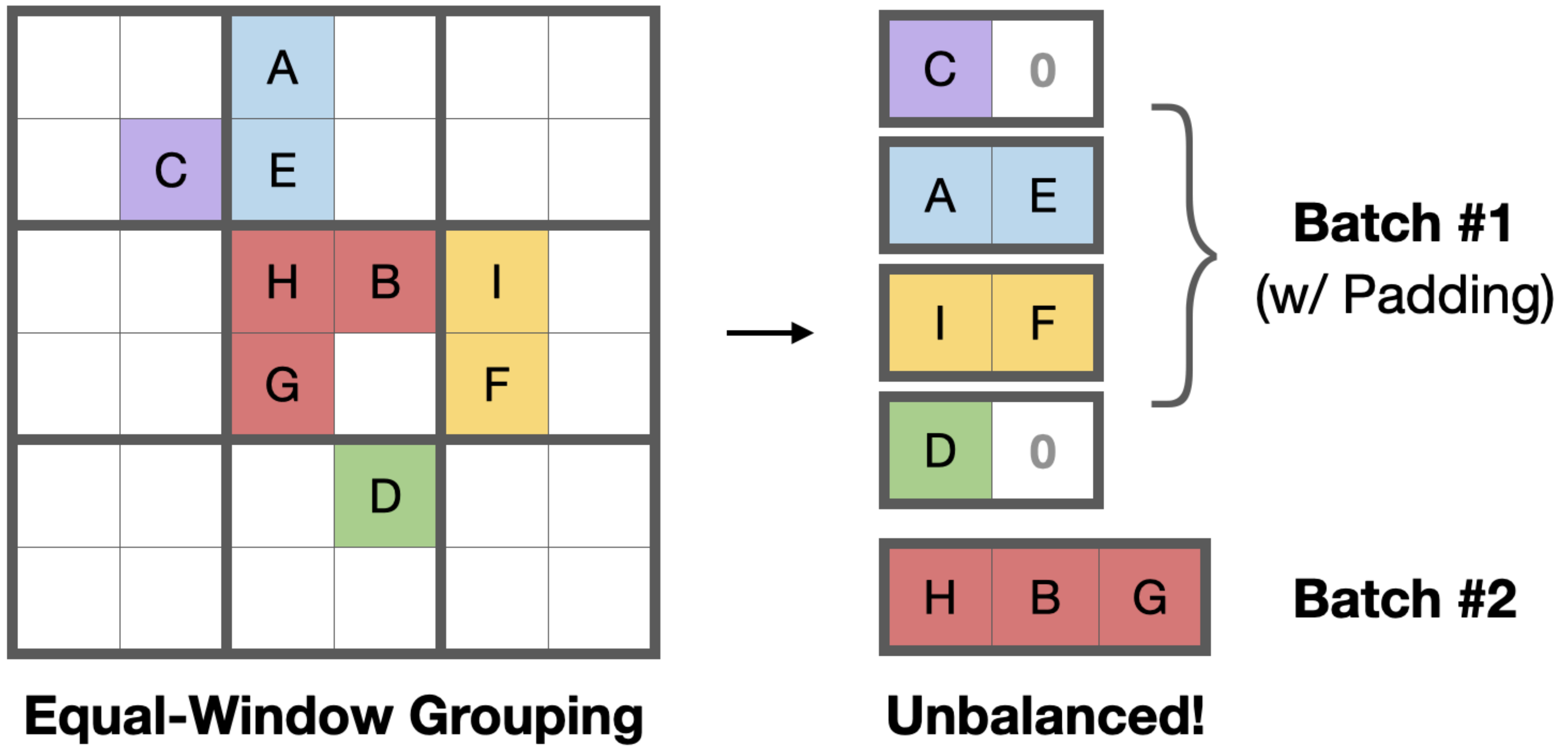
Window PCTs suffer from the **padding and partitioning overhead** due to the unbalanced workload across windows.

Fan et al., "Embracing Single Stride 3D Object Detector with Sparse Transformer", CVPR 2022

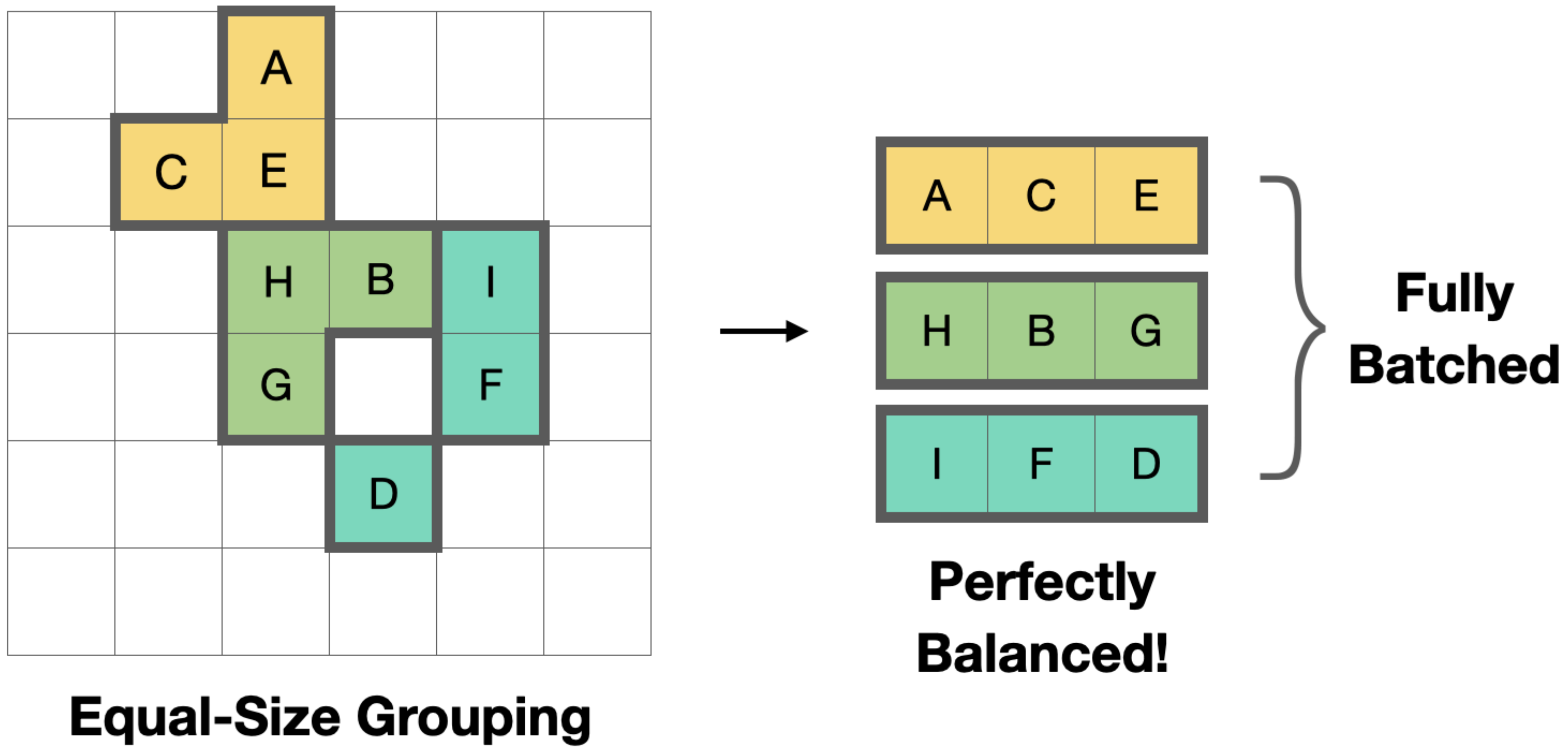
FlatFormer: Flattened Window Attention

Equal-size grouping trades spatial proximity for computational regularity

Equal-window grouping maintains perfect **spatial proximity** but breaks the **computational regularity**.

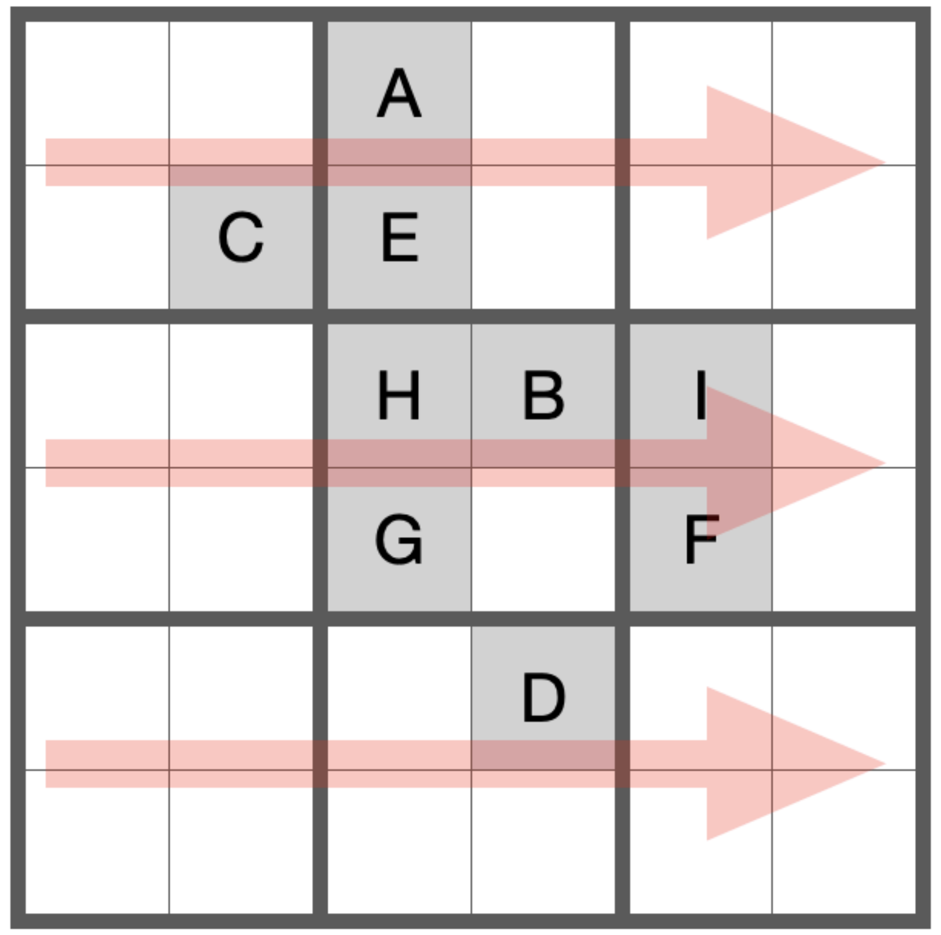


Equal-size grouping ensures balanced **computation workload** but cannot guarantee the **geometric locality**.

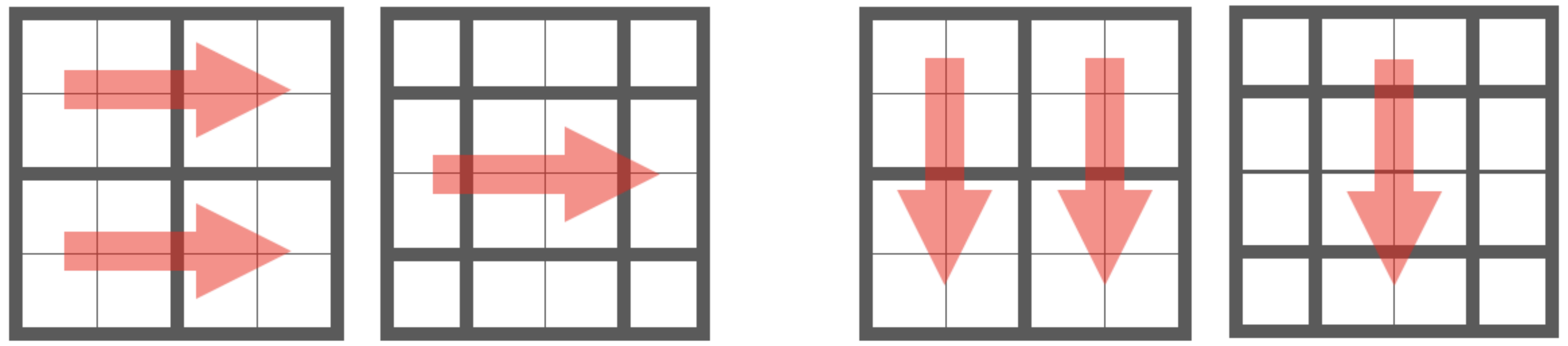


FlatFormer: Flattened Window Attention

Window-based sorting preserves geometric locality after flattening



Sort all points first by **window coordinates** and then by **local coordinates** within the window.

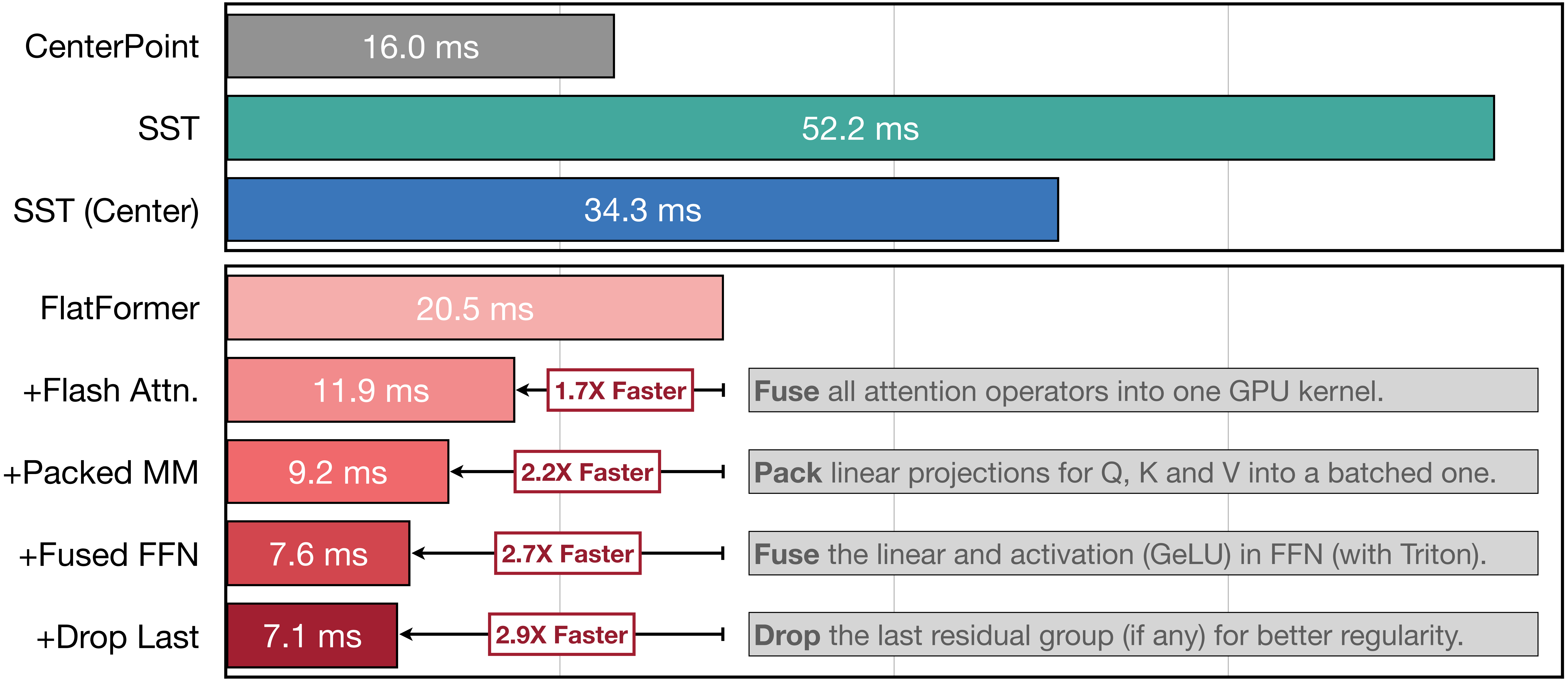


Window Shift **Alternate Sorting Axis**

Sorting Criterion	Window Shifting	Axis Alternation	Mean L2 mAPH
Random	—	—	57.8
Point	—	Y	60.4
Window	Y	**N**	61.1
Window	**N**	Y	61.2
Window	**Y**	**Y**	**61.7**

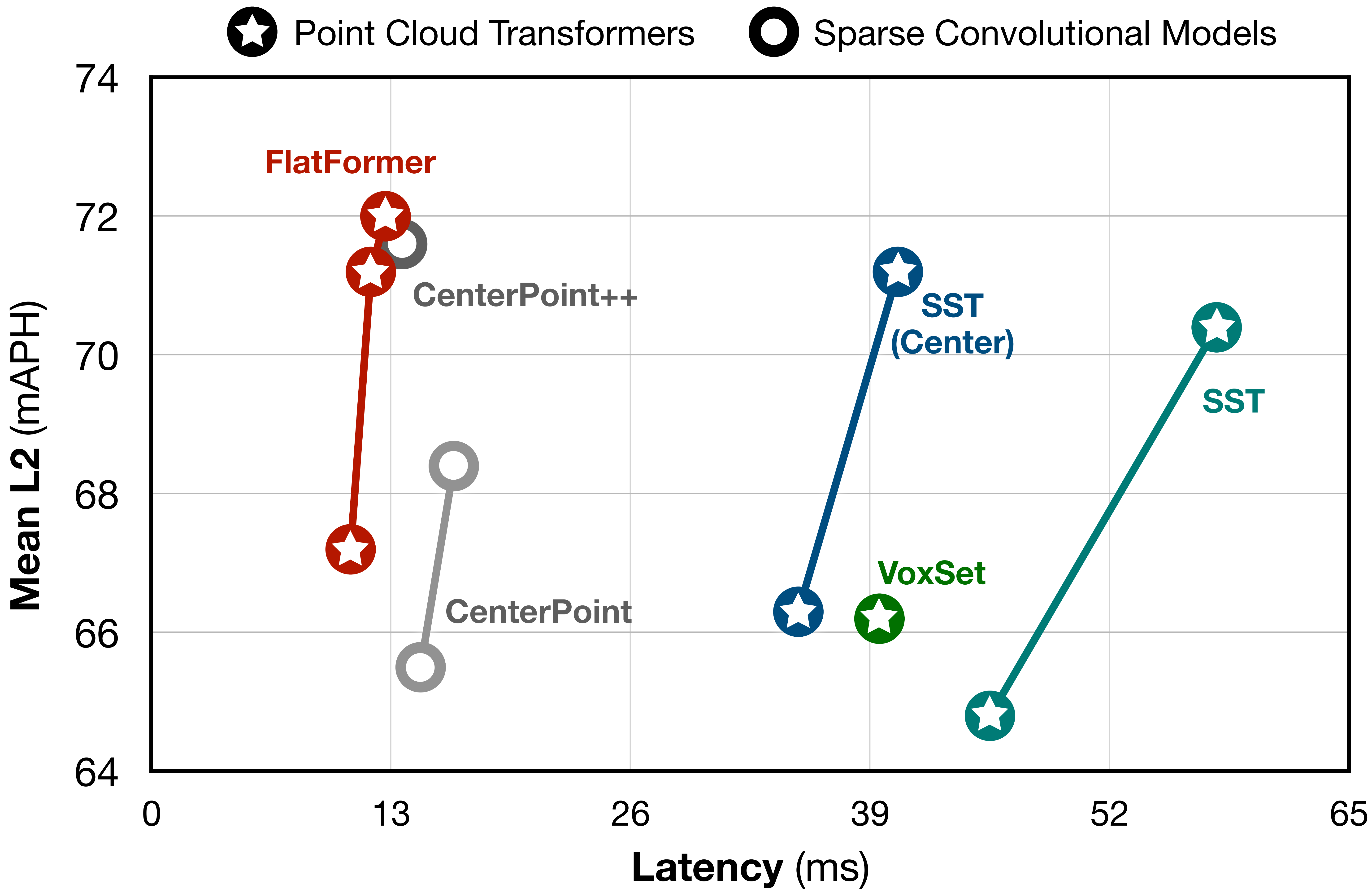
Efficient Implementation

FlatFormer can benefit from existing system optimizations for transformers



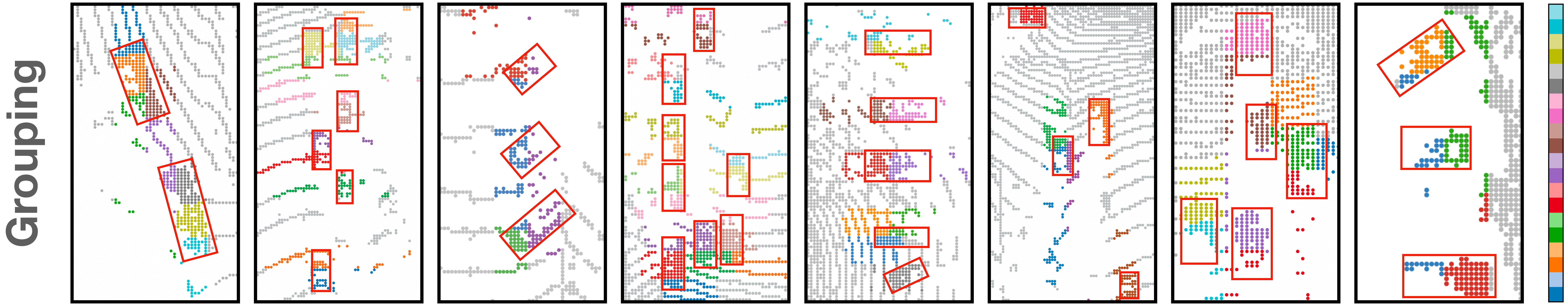
Results: 3D Object Detection on Waymo

FlatFormer closes the latency gap between PCTs and SpConv-based models

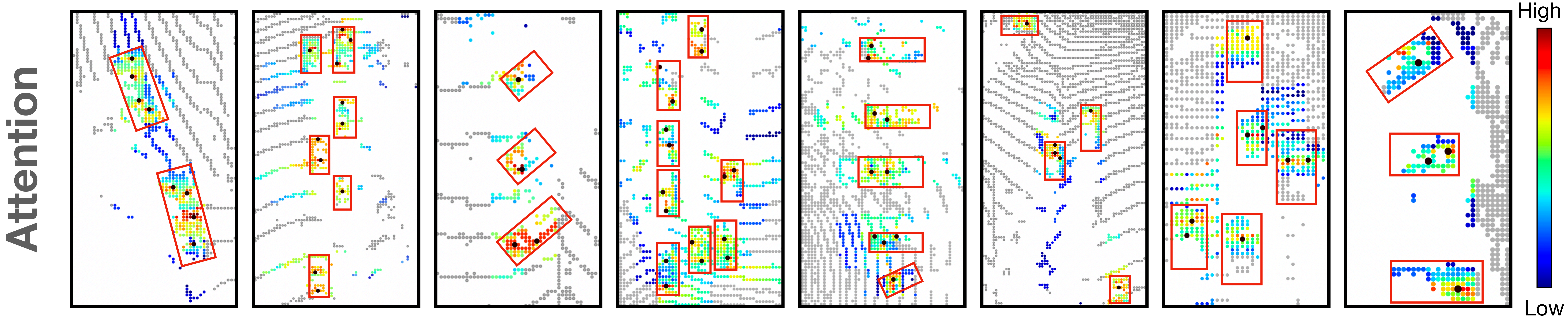


Analysis: Grouping & Attention

Equal-size grouping is **mostly spatially local**.



Attention learns to **suppress outlier points**.



Deployment on Edge GPUs

First point cloud transformer that achieves real-time performance on edge GPUs!