Polar Hierarchical Mamba: Towards Streaming LiDAR Object Detection with Point Clouds as Egocentric Sequences

Mellon M. Zhang* Glen Chou
Georgia Institute of Technology
{meilongz, chou} @ gatech.edu

Abstract

*Accurate and efficient object detection is a crucial component for fully autonomous self-driving. LiDAR sensors are employed to augment or replace cameras for more robustness in diverse driving situations, making object detection on LiDAR point clouds a critical area of research and improvement. Traditional approaches to LiDAR object detection wait for a full 360 degree turn of the scanning sensor before processing the entire point cloud in one go, introducing significant latency and lowering throughput. Previous streaming approaches use the raw LiDAR polar coordinate system to process egocentric partial scans of point clouds, but rely on translation-invariant convolutions, which are incompatible with polar coordinates and lead to performance degradation. In this paper, we show that the reliance on convolutions is not necessary and propose a Mamba-only backbone with **P**olar **Hi**erarchical **M**amba (PHiM) blocks, aggregating per-point features within each partial scan with a local bidirectional state space model and capturing higher-level global features in a streaming fashion with a global forward state space model. Our model on the Waymo Open dataset demonstrates 10% performance improvement from the previous leading polar-based detector, featuring state of the art performance among all polar-based methods while being competitive with existing Cartesian-based detectors with a 2x improvement in processing throughput evaluated as predictions per second.*

1. Introduction

Object detection is a vital task for autonomous vehicles (AVs), requiring not only high accuracy but also robustness and efficiency due to the unpredictability of real-world driving. To enhance perception, most AVs incorporate LiDAR sensors, which outperform other modalities in low-light and long-range scenarios. Traditional LiDAR methods [8, 21, 34, 37, 46, 48, 57] process full or aggregated point cloud scans [2, 4, 38] which leads to significant latency— hundreds of milliseconds—making real-time detection challenging. This has sparked interest in streaming approaches that process partial scans directly [5, 9, 15, 31].

Polar coordinates provide an efficient format for streaming LiDAR data but introduce spatial distortion, making them incompatible with standard convolutional backbones [31]. Prior methods [5, 31, 45] attempt to correct for these distortions post hoc, often resulting in complex architectures. In contrast, we propose a simple yet effective solution using the Mamba state space model [13], which avoids convolution-heavy processing and does not rely on translation invariance. Mamba enables near-linear sequence modeling and has shown strong performance across modalities [7, 18, 61]. By treating streaming LiDAR data as egocentric sequences of partial scans, we introduce a Polar Hierarchical Mamba (PHiM) architecture that processes each scan independently and efficiently, without the need for serialization techniques or positional encodings. Our approach achieves state-of-the-art results among streaming methods, competitive accuracy compared to Cartesian-based models, and supports fast, pipelined inference—all within a clean and streamlined architecture. Our contributions are summarized as follows:

- We introduce a Polar Hierarchical Mamba (PHiM) block, which enables fast local and global feature learning on egocentric sequences of point cloud sectors.
- We introduce decomposed dimensional convolutions to downsample sparse 3D feature maps while avoiding the polar-distorted (r, θ) plane, enabling application of polar-based methods for other 3D LiDAR tasks.
- We provide a thorough comparison of our method against existing Cartesian and polar-based detectors on the Waymo Open dataset, demonstrating new state of the art performance among all polar methods and competitive performance among Cartesian methods with a fraction of the end-to-end latency.

2. Related work

Convolution-based methods. Early LiDAR object detection methods borrowed heavily from image-based detection, transforming raw point clouds into structured voxel grids and applying dense 3D convolutions for feature learning. VoxelNet [57] exemplifies this approach by defining a 3D voxel grid and using PointNet [34] to extract features within each voxel. To reduce

*Corresponding Author

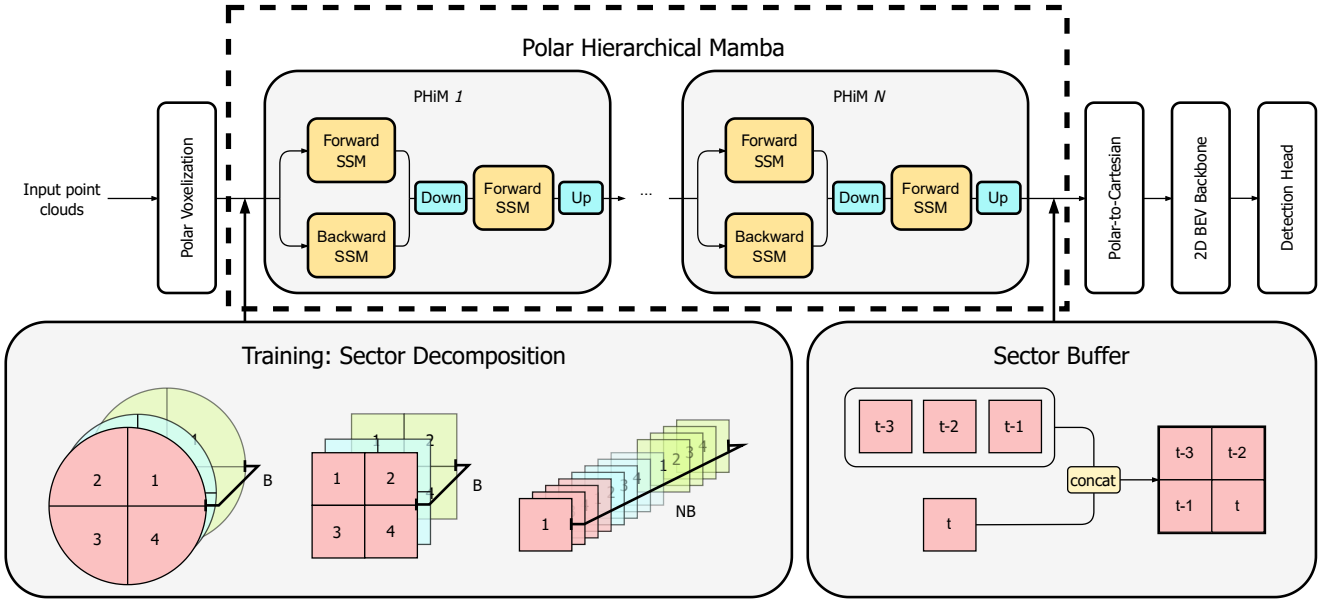

Figure 1. Our model pipeline begins by voxelizing input point clouds into polar-based voxel grids. To simulate a streaming setup, the resulting feature map is divided into individual sectors. Within each sector, stacked PHiM blocks (see Fig. 2) perform three key operations: they aggregate local features, encode sector-level representations, and propagate global information across sectors in the forward (temporal) direction to maintain causality. The output is a 3D feature map, which is then concatenated with buffered features from previous sectors. This combined representation is condensed into a 2D bird's-eye-view (BEV) format and transformed into the Cartesian plane. A BEV convolutional backbone further processes the features, and predictions are made using a CenterPoint-based detection head. Our main contributions are highlighted within the dashed box in the model overview.

the high computational cost of dense 3D processing, pillar-based methods [21, 23, 36] project the 3D data into 2D bird's-eye view (BEV) maps and employ 2D convolutions, trading some accuracy for efficiency. SECOND [46] marked a major shift by introducing sparse [11] and submanifold [12] convolutions, enabling efficient, direct processing of sparse features and paving the way for hybrid 3D-2D backbones that balance geometric richness and computational efficiency. More recently, CenterPoint [48] addressed spatial misalignment in bounding box representations by detecting objects as points. This idea has since been extended by multiple models [1, 17, 51, 52, 59]. Following this line, our method also adopts a CenterPoint detection head to mitigate spatial misalignment issues caused by polar representations.

Non-convolution methods. Transformers [26, 41] and Mamba [7, 13] have gained traction in LiDAR tasks [1, 24, 25, 27, 30, 33, 39, 42, 47, 56, 59, 60], building on their success in language modeling. SWFormer [39] organizes point clouds into pillars and applies local 2D window attention, while DSVT-Voxel [42] uses voxel-based formats with grouped set attention to better capture 3D structure. LION [28] showed linear RNNs can achieve state-of-the-art performance, sparking interest in linear models. Voxel Mamba [53] leverages Hilbert curves [16] to serialize voxelized point clouds into 1D sequences for a group-free state space model, capturing spatial locality through multi-granularity implicit window embeddings. UniMamba [19] uses Z-curves [32] to form local patches that aggregate both local and global spatial context. Numerous other works [10, 22, 29, 43, 44, 49, 50, 54] have extended Mamba to various point cloud tasks. However, existing approaches primarily serialize point clouds spatially and overlook the temporal dimension inherent to scanning LiDAR sensors. Consequently, egocentric sequences of partial LiDAR sectors remain underexplored. In contrast, our method is the first to explicitly leverage the temporal dimension by treating point clouds as egocentric sequences of sectors. This eliminates the need for complex serialization, spatial windowing, or positional encodings.

Polar coordinates. Polar coordinate representations of LiDAR point clouds have gained traction due to their uniform point distribution across polar voxels [31] and their alignment with the native scanning format of LiDAR sensors [5], making them ideal for efficient streaming perception pipelines. Polar-based methods have shown strong performance—sometimes surpassing Cartesian approaches—in tasks like semantic segmentation [55] and semantic occupancy prediction [45, 62]. Motivated by these results, several efforts [3, 5, 31, 35] have attempted to apply polar coordinates to object detection. However, object detection demands precise structural understanding, and polar representations inherently introduce distortions—e.g., the distance between two azimuth angles increases with radial distance—leading to misalignment with translation-invariant

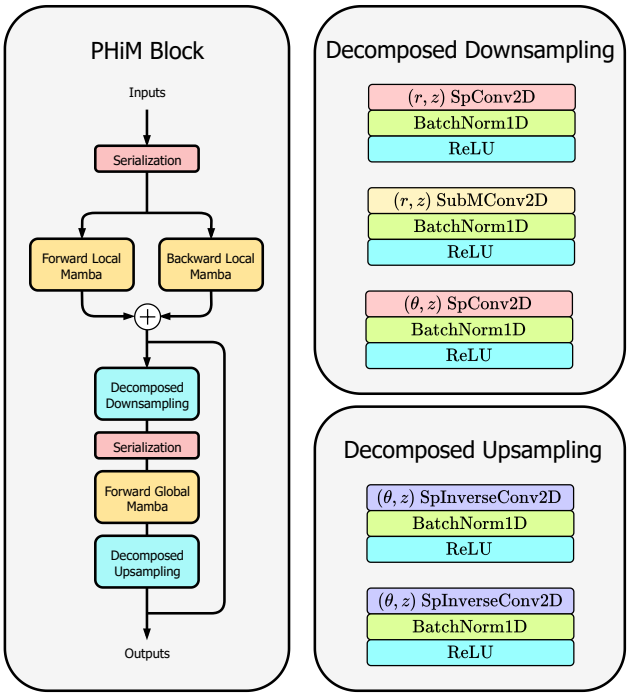

Figure 2. (Left) The Polar Hierarchical Mamba (PHiM) block. Serialization is according to the azimuth angle, and the bidirectional local SSM is aggregated with an elementwise addition. (Right) Decomposed downsampling and upsampling operations.

convolutions. Prior work tackled this with convolutional backbones and custom distortion-mitigation strategies, which either led to inferior performance [5] or overly complex models [31]. Our core hypothesis is that Mamba [13], a state space model that does not rely on translation invariance, can overcome the distortion issues of polar representations. This allows us to build a streamlined pipeline that retains structural accuracy and delivers competitive performance.

3. Method

Figure 1 shows an overview of our architecture. Like prior methods, we use a voxel feature encoder to convert partial scans into sparse 3D features, modified to operate in polar rather than Cartesian coordinates. These features are processed by a state space model (SSM)-based (Sec. 3.1) 3D backbone composed of stacked Polar Hierarchical Mamba (PHiM) blocks (Sec. 3.2), which aggregate local spatial information bidirectionally. Voxel-wise features are then converted into sector-wise representations using decomposed convolutions (Sec. 3.3). A global state space model aggregates features across sectors using historical context. These sector features are concatenated with past features to form a complete point cloud feature map, which is transformed into a Cartesian bird's-eye view (BEV) by compressing the height dimension (Sec. 3.4). A 2D convolutional backbone and CenterPoint-based detection head perform final prediction. For efficient training, we split full point clouds into sectors and enforce temporal causality by preventing token mixing across time steps.

Algorithm 1: PHiM Block Forward Pass

Data: Voxel features V, coordinates C, batch size B, spatial shape S
Result: Updated features and coordinates
Partition space into F frustums along azimuth; update C to encode frustum IDs;
Adjust S and B to reflect frustum partitioning;
Initialize local feature tensor V_{local};
foreach *frustum index* f **do**
  Extract V_f from V;
  $F_{\text{fw}} \leftarrow \texttt{fsm}(V_f)$;
  $F_{\text{bw}} \leftarrow \texttt{bsm}(V_f^{\text{rev}})$;
  $F_{\text{comb}} \leftarrow F_{\text{fw}} + F_{\text{bw}}^{\text{rev}}$;
  Assign F_{comb} to V_{local} at frustum f;
end
Construct SparseConvTensor X from V_{local},C;
foreach *encoder* E **do**
  $X \leftarrow E(X)$;
end
Decode frustum IDs in C to restore original layout;
Initialize global feature tensor V_{global};
foreach *batch index* b **do**
  Extract F_b from X;
  $F_g \leftarrow \texttt{fgm}(F_b)$;
  Assign F_g to V_{global} for batch b;
end
Update $X.features \leftarrow \texttt{normalize}(V_{\text{global}})$;
Re-encode frustum IDs and update S, B accordingly;
foreach *decoder* D **do**
  $X \leftarrow D(X)$;
end
Apply residual: $X.features \leftarrow X.features + V_{\text{local}}$;
Decode frustum IDs in C to obtain final layout;
return $X.features$, $X.indices$;

3.1. Background

The state space (SSM) model [14] continuous system maps a 1D input $x(t) \in \mathbb{R}^L$ to an output signal $y(t) \in \mathbb{R}^L$ via a hidden state $h(t) \in \mathbb{R}^N$. This can be represented as the following set of linear differential equations:

$$\begin{cases} h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) = \mathbf{C}h(t) + \mathbf{D}x(t), \end{cases} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times N}$, $\mathbf{C} \in \mathbb{R}^{1 \times N}$ are the learnable parameters and $\mathbf{D} \in \mathbb{R}^{\mathbf{1}}$ is a residual connection. Mamba [13] discretizes the SSM model parameters $\mathbf{A}$ and $\mathbf{B}$ using the zero-order hold (ZOH) transformation and a timescale parameter

Type	Method	mAP/mAPH		Vehicle AP/APH		Pedestrian AP/APH		Cyclist AP/APH	
		L1	L2	L1	L2	L1	L2	L1	L2
Cartesian	SECOND [46]	67.2/63.1	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
	PointPillar [21]	69.0/63.5	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
	CenterPoint [48]	75.9/73.5	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
	DSVT-Voxel [42]	80.3/78.2	74.0/72.1	79.7/79.3	71.4/71.0	83.7/78.9	76.1/71.5	77.5/76.5	74.6/73.7
	HEDNet [51]	81.4/79.4	75.3/73.4	81.1/80.6	73.2/72.7	84.4/80.0	76.8/72.6	78.7/77.7	75.8/74.9
	VoxelNeXt [6]	78.6/76.3	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
	Voxel Mamba [53]	-/79.6	-/73.6	80.8/80.3	72.6/72.2	85.0/80.8	77.7/73.6	78.6/77.6	75.7/74.8
	UniMamba [19]	-/-	76.1/74.1	80.6/80.1	72.3/71.8	86.0/81.3	78.7/74.1	80.3/79.3	77.5/76.5
Polar	PolarStream [5, 31]*	-/-	-/60.9	72.4/71.8	64.6/64.0	-/-	-/-	-/-	-/-
	PARTNER [31]	-/-	-/63.1	77.8/77.2	70.3/69.8	-/-	-/-	-/-	-/-
	PHiM (ours)	78.5/76.6	**72.1/70.3**	**79.2/78.6**	**71.0/70.5**	80.7/76.6	72.7/68.8	75.5/74.4	72.7/71.7

Table 1. Comparison with prior methods on the Waymo Open validation set. Metrics: mAP/mAPH (%)↑ for overall results, AP/APH (%)↑ for each category. * denotes reimplementation results.

Δ, where $\overline{\mathbf{A}} = \exp(\mathbf{\Delta A})$, $\overline{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta A}$. The resulting discretized equations are as follows:

$$\begin{cases} h_t = \mathbf{A} h_{t-1} + \mathbf{B} x_t, \\ y_t = \mathbf{C} h_t, \end{cases} \tag{2}$$

Finally, the models compute output through an efficient reformulation as a global convolution:

$$\begin{cases} \overline{\mathbf{K}} = (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}\mathbf{B}}, ..., \mathbf{C}\overline{\mathbf{A}}^{\mathbf{k}}\overline{\mathbf{B}}), \\ \mathbf{y} = \mathbf{x} * \overline{\mathbf{K}} \end{cases} \tag{3}$$

Mamba combines the time-varying strength of self-attention, near-linear scaling of recurrent neural networks and fast training of convolutions for efficient modeling of sequences.

3.2. PHiM block

The left side of Fig. 2 shows the PHiM block. Sparse input features are first serialized by azimuth, radial distance, and height—directly given by voxel indices in polar coordinates. Two Mamba blocks then process the features in forward and backward directions, and their outputs are summed to enhance local spatial representation within each sector. To capture global context, voxel features are converted into high-level sector features using decomposed convolutions (Sec. 3.3), avoiding the distortion-prone (r,θ) plane. These sector features are then re-serialized and passed into a global Mamba block, which aggregates information from previous sectors through its hidden state. Pseudocode of the PHiM block forward pass is shown in Alg. 1.

3.3. Decomposed convolutions

The right side of Fig. 2 illustrates our decomposed downsampling and upsampling strategy. Standard convolutions struggle on polar feature maps due to positional variance—objects can appear distorted depending on location—leading to degraded performance. Yet, convolutions remain efficient and effective, particularly for resolution changes. To retain their benefits while mitigating distortion, we decompose 3D convolutions into 2D ones. Since most distortion occurs in the (r,θ) plane—whereas (r,z) forms a vertical cross-section and θ doesn't scale with z—we avoid direct convolution in (r,θ). Instead, we apply 2D convolutions on the (r,z) and (θ,z) planes. Downsampling uses a sparse 2D convolution with stride $(3,3)$ on (z,r), followed by a submanifold 2D convolution with the same stride, then a sparse convolution with stride $(1,3)$ on (z,θ). Upsampling involves inverse convolutions with the same stride parameters but in reverse. The third axis is reshaped into a batch dimension, enabling independent processing. This decomposition decouples the scaling effects of r and θ, allowing convolution kernels to better generalize in more translation-invariant planes.

3.4. Polar to Cartesian mapping

To enable efficient downstream processing with convolution-based bird's-eye-view (BEV) backbones, we apply a polar-to-Cartesian mapping along with height compression. Specifically, we compute Cartesian indices for each sparse feature and average features within the same bin, converting 3D PHiM outputs into a format optimized for 2D convolutional backbones.

3.5. Training and implementation

We use the open-souce OpenPCDet [40] toolbox to implement our method. During training, we split full point clouds into sectors, treating each sector as an individual batch sample to prevent any token-mixing between time steps. This process is illustrated at the bottom left of Fig. 1.

4. Results

Comparison with previous methods. We use reported results from original papers in our table and scatterplot. For our latency evaluations, we use specified config files from the OpenPCDet [40] codebase for the Cartesian based methods and reported

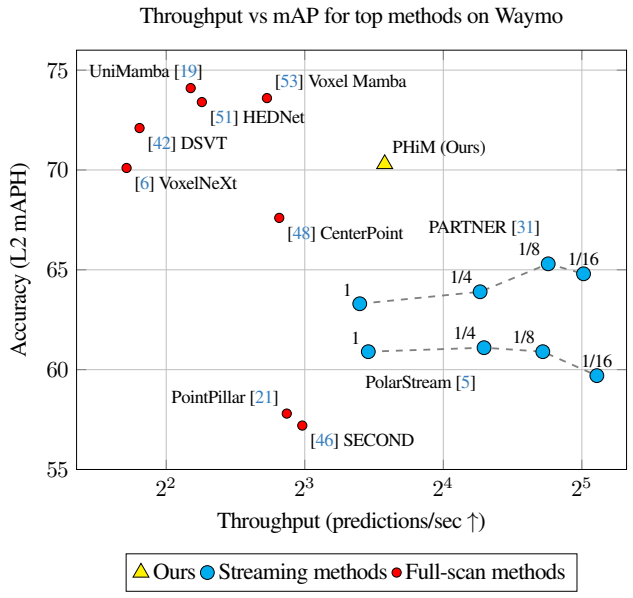

Figure 3. Throughput vs performance on Waymo Open.

Streaming	DDC	PHiM	Polar2Cart	L1 mAP
✗	✗	✗	✗	13.13
✗	✓	✗	✗	16.33
✗	✓	✗	✓	22.00
✓	✗	✓	✗	21.97
✓	✗	✓	✓	26.01
✓	✓	✓	✗	25.87
✓	✓	✓	✓	41.20

Table 2. Waymo L1 mAP for different combinations of decomposed depthwise convolutions (DDC), polar hierarchical mamba (PHiM) blocks and polar to cartesian mapping (Polar2Cart). Streaming indicates whether the model supports processing on partial LiDAR sectors.

results from the original polar-based method papers. All mAP results are on the validation sets. For comparisons with Cartesian based methods, we evaluate our model on length 4 sequences of 1/4 LiDAR sectors - equivalent to one full point cloud.

Measuring throughput. In a dynamic world with sudden unexpected movements, perception models must be able to quickly react. Traditional full-scan methods are unable to react fast enough because they are limited by the artificial sensor latency of one full 360 degree LiDAR scan. This motivates evaluation of the throughput - that is, how many predictions or forward passes of the model is able to be achieved in one second. Higher throughput indicates a faster reaction speed to the dynamic world, potentially increasing robustness. To this end, we measure the end-to-end model latency for each model as the mean CUDA walltimes over 1000 samples. We measure this latency on batch size 1 input on 1 H200 GPU. We add on calculated sensor latency, where one full 360 degree LiDAR scan is estimated to take 100 ms - in line with the Velodyne HDL-64E scanning LiDAR sensor. We then compute the throughput as model inferences per second.

Results on Waymo Open. Table 1 shows a comparison of Polar Hierarchical Mamba with existing state of the art methods on the Waymo Open [38] validation set under the full-scan setting. For these evaluations, we take the streaming polar methods and pass in an entire point cloud as the sole input sector. Notably, our method presents a 10% performance increase over the previous leading polar method PARTNER [31] and showcases competitive performance against leading Cartesian methods.

Performance evaluation. Figure 3 compares performance and throughput (inferences per second) for leading methods on the Waymo Open validation set. We compare against full-scan (shown in red) and streaming methods (shown in blue). Notably, full-scan methods are limited by the artificial latency introduced by the sensor scanning process, and thus have a theoretical maximum throughput of 10 predictions per second. Streaming methods face lower and lower latency limits as the size of each individual sector decreases, and are able to achieve much higher throughput. We report our results on 1/4 LiDAR sectors, showing 2x throughput compared to methods of similar performance.

Ablation study. Table 2 ablates our decomposed convolutions (DDC), polar hierarchical mamba (PHiM) block and polar to cartesian (polar2cart) mapping. We train each method on a fixed 1/100 subsample of the Waymo Open dataset and report the L1 mean average precision on the validation set. Importantly, all three components are necessary for maximum performance: decomposed convolutions to capture spatial information without distortion warp, hierarchical mamba is needed to aggregate local and global scale information, and polar-to-cartesian mapping is needed to allow compatibility with downstream 2D convolutions and the Centerpoint [48] head.

5. Conclusion

We present Polar Hierarchical Mamba (PHiM), a simple and stackable polar-based architecture which efficiently processes LiDAR point clouds as an egocentric sequence of partial sectors. On the Waymo Open dataset, PHiM demonstrates state of the art performance compared to previous polar methods and competitive performance with Cartesian methods with 2x throughput.

Acknowledgements

This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022. 2

[2] Wilson Benjamin, Qi William, Agarwal Tanmay, Lambert John, Singh Jagjeet, Khandelwal Siddhesh, Pan Bowen, Kumar Ratnesh, Hartnett Andrew, Kaesemodel-Pontes Jhony, Ramanan Deva, Carr Peter, and Hays James. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 1

[3] Manoj Bhat, Steve Han, and Fatih Porikli. Fast Polar Attentive 3D Object Detection on LiDAR Point Clouds. 2

[4] 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*, 2020. 1

[5] Qi Chen, Sourabh Vora, and Oscar Beijbom. Polarstream: Streaming object detection and segmentation with polar pillars. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2, 3, 4, 5

[6] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, 2023. 4, 5

[7] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024. 1, 2

[8] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully Sparse 3D Object Detection. In *NeurIPS*, 2022. 1

[9] Davi Frossard, Simon Suo, Sergio Casas, James Tu, Rui Hu, and Raquel Urtasun. Strobe: Streaming object detection from lidar packets, 2020. 1

[10] Youjia Fu, Zihao Xu, Junsong Fu, Huixia Xue, Shuqiu Tan, Lei Li, and Shaoxun Qing. Monomm: a multi-scale mamba-enhanced network for real-time monocular 3d object detection. *The Journal of Supercomputing*, 81(3):449, 2025. 2

[11] Ben Graham. Sparse 3d convolutional neural networks. *arXiv preprint arXiv:1505.02890*, 2015. 2

[12] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2

[13] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 2, 3

[14] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. 3

[15] Wei Han, Zhengdong Zhang, Benjamin Caine, Brandon Yang, Christoph Sprunk, Ouais Alsharif, Jiquan Ngiam, Vijay Vasudevan, Jonathon Shlens, and Zhifeng Chen. Streaming object detection for 3-d point clouds, 2020. 1

[16] David Hilbert. *Ueber die stetige Abbildung einer Linie auf ein Flächenstück*. 2

[17] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *AAAI*, 2022. 2

[18] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024. 1

[19] Xin Jin, Haisheng Su, Kai Liu, Cong Ma, Wei Wu, Fei Hui, and Junchi Yan. Unimamba: Unified spatial-channel representation learning with group-efficient mamba for lidar-based 3d object detection, 2025. 2, 4, 5

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[21] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1, 2, 4, 5

[22] Heng Li, Yuenan Hou, Xiaohan Xing, Xiao Sun, and Yanyong Zhang. Occmamba: Semantic occupancy prediction with state space models. *arXiv preprint arXiv:2408.09859*, 2024. 2

[23] Jinyu Li, Chenxu Luo, and Xiaodong Yang. Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. In *CVPR*, 2023. 2

[24] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. In *NIPS*, 2022. 2

[25] Yuhang Liu, Yinji Ge, Mengyue Li, Guixu Zheng, Boyi Sun, and Fei-Yue Wang. Pillarmamba: A lightweight mamba-based model for 3d object detection. In *2024 IEEE 4th International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pages 652–655, 2024. 2

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[27] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *CVPR*, 2023. 2

[28] Zhe Liu, Jinghua Hou, Xinyu Wang, Xiaoqing Ye, Jingdong Wang, Hengshuang Zhao, and Xiang Bai. Lion: Linear group rnn for 3d object detection in point clouds, 2024. 2

[29] Dening Lu, Linlin Xu, Jun Zhou, Kyle Gao, Zheng Gong, and Dedong Zhang. 3d-umamba: 3d u-net with state space model for semantic segmentation of multi-source lidar point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 136:104401, 2025. 2

[30] Jiageng Mao, Yujing Xue, Minzhe Niu, et al. Voxel transformer for 3d object detection. *ICCV*, 2021. 2

[31] Ming Nie, Yujing Xue, Chunwei Wang, Chaoqiang Ye, Hang Xu, Xinge Zhu, Qingqiu Huang, Michael Bi Mi, Xinchao Wang, and Li Zhang. Partner: Level up the polar representation for lidar 3d object detection, 2023. 1, 2, 3, 4, 5

[32] Jack A. Orenstein. Spatial query processing in an object-oriented database system. In *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, page 326–336, New York, NY, USA, 1986. Association for Computing Machinery. 2

[33] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, 2022. 2

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1

[35] Meytal Rapoport-Lavie and Dan Raviv. It's all around you: Range-guided cylindrical network for 3d object detection, 2020. 2

[36] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *ECCV*, 2022. 2

[37] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 5

[39] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *ECCV*, 2022. 2

[40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 4, 1

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2

[42] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023. 2, 4, 5

[43] Tao Wang, Wei Wen, Jingzhi Zhai, Kang Xu, and Haoming Luo. Serialized point mamba: A serialized point cloud mamba segmentation model. *arXiv preprint arXiv:2407.12319*, 2024. 2

[44] Zicheng Wang, Zhenghao Chen, Yiming Wu, Zhen Zhao, Luping Zhou, and Dong Xu. Pointramba: A hybrid transformer-mamba framework for point cloud analysis. *arXiv preprint arXiv:2405.15463*, 2024. 2

[45] Yujing Xue, Jiaxiang Liu, Jiawei Du, and Joey Tianyi Zhou. Pvp: Polar representation boost for 3d semantic occupancy prediction, 2024. 1, 2

[46] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. In *Sensors*, 2018. 1, 2, 4, 5

[47] Mao Ye, Gregory P. Meyer, Yuning Chai, and Qiang Liu. Efficient transformer-based 3d object detection with dynamic token halting. In *ICCV*, 2023. 2

[48] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 1, 2, 4, 5

[49] Zihan You, Hao Wang, Qichao Zhao, and Jinxiang Wang. Mambabev: An efficient 3d detection model with mamba2, 2024. 2

[50] Kang Zeng, Hao Shi, Jiacheng Lin, Siyu Li, Jintao Cheng, Kaiwei Wang, Zhiyong Li, and Kailun Yang. Mambamos: Lidar-based 3d moving object segmentation with motion-aware state space model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1505–1513, 2024. 2

[51] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li Li, and Xiaolin Hu. HEDNet: A hierarchical encoder-decoder network for 3d object detection in point clouds. In *NeurIPS*, 2023. 2, 4, 5

[52] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, Si Liu, and Xiaolin Hu. SAFDNet: A simple and effective network for fully sparse 3d object detection. In *CVPR*, 2024. 2

[53] Guowen Zhang, Lue Fan, Chenhang He, Zhen Lei, Zhaoxiang Zhang, and Lei Zhang. Voxel mamba: Group-free state space models for point cloud based 3d object detection. *arXiv preprint arXiv:2406.10700*, 2024. 2, 4, 5, 1

[54] Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024. 2

[55] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation, 2020. 2

[56] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. Octr: Octree-based transformer for 3d object detection. In *CVPR*, 2023. 2

[57] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1

[58] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *CoRL*, 2019. 1

[59] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *ECCV*, 2022. 2

[60] Benjin Zhu, Zhe Wang, Shaoshuai Shi, Hang Xu, Lanqing Hong, and Hongsheng Li. Conquer: Query contrast voxel-detr for 3d object detection, 2022. 2

[61] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024. 1

[62] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. 2