2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

VeloVox: A Low-Cost and Accurate 4D Object Detector with Single-Frame Point Cloud of Livox LiDAR

Tao Ma[1,2*], Zhiwei Zheng[3,2*], Hongbin Zhou[2], Xinyu Cai[2], Xuemeng Yang[2], Yikang Li[2], Botian Shi[2] and Hongsheng Li[1]

***Abstract*— Combining motion prediction in LiDAR-based 3D object detection is an effective method for improving overall accuracy, especially the downstream autonomous driving tasks. The recent development of low-cost LiDARs (e.g. Livox LiDAR) enables us to explore such 4D perception systems with a lower budget and higher performance. In this paper, we propose a 4D object detector, VeloVox, to establish accurate object detection and velocity estimation with a single-frame point cloud of Livox LiDAR. Based on the non-repetitive scanning pattern and point-level temporal nature, we propose a two-stage module to enhance the spatial-temporal point feature interaction along the time dimension. The aggregated feature also benefits a more accurate proposal refinement. To demonstrate the performance, comparison of VeloVox with several SOTA detector based baselines is evaluated on our in-house dataset and synthesized dataset built under Carla simulation. Code will be released at https://github.com/PJLab-ADG/VeloVox.**

I. INTRODUCTION

Light Detection and Ranging (LiDAR), as a high-end depth sensor, provides accurate measurements and system robustness, and has been widely used to accomplish several perception tasks, such as 3D object detection [1]–[5], segmentation [6], and object tracking [7]–[10]. Current 3D object detectors heavily rely on the dense point cloud to capture effective feature representation for determining the location and dimension of potential objects [11]–[13]. For example, concatenating multiple frames of low-beam point clouds [14] is widely used to approximate the high-beam LiDARs. In addition to benefiting for predicting motion information, it also introduces serious point cloud blurring problems, which further degrades the performance and brings high computation overheads [15], [16].

As the autonomous industry progresses, many new technology developments have enabled the commercialization of low-cost LiDARs, e.g. Ouster and Livox LiDARs. As shown in Fig. 1 (a), featuring a non-repetitive scanning pattern, Livox LiDARs pose unique advantages in low-cost LiDAR-assisted perception systems: (1) sufficient amount of high-density point cloud data for 3D detection shown in Fig. 2, and (2) point-level temporal information to predict the motion of objects. Besides, this special scanning pattern results in the tailing phenomenon as Fig. 1 (c) shows. However, directly applying standard 3D detectors with voxel- or pillar-based griding schemes on multi-frame input is non-trivial.

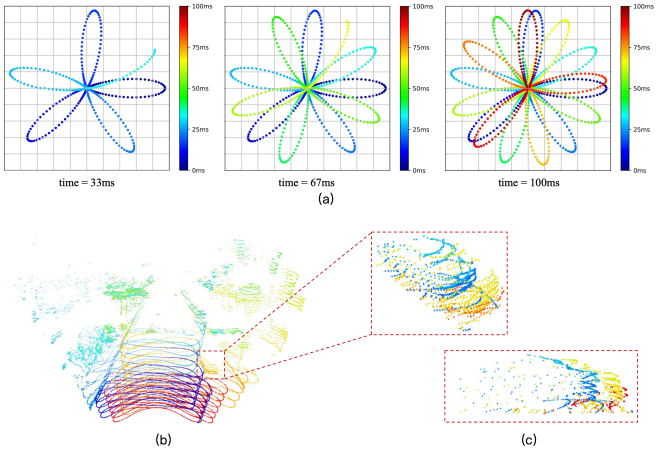

Fig. 1: (a) The non-repetitive scanning pattern of 3D points projected on the plane of 1m distance in front, where the color encodes the sampling time. (b) A frame of Livox point cloud in the surroundings. (c) The long-tailing phenomenon of a dynamic car, where the object points are colored based on the corresponding timestamps.

To interact safely and effectively with other road participants, autonomous vehicles must accurately predict the velocities and activities of surrounding vehicles. Current 3D detectors usually take as input the concatenation of multiple frames of point clouds, and utilize another parallel head to tackle the velocity estimation. All the points in each frame are decorated with the same time offset to distinguish different frames, which is considering as an additional attribute. They are further processed with PointNet-based networks to extract features together with the spatial coordinates, causing the difference between the spatial and temporal features to be ignored. In addition, the points divided into the same voxels or pillars are always average-pooled, which tries to reduce the adverse effects of the tailing phenomenon [16]–[18]. But this unique characteristic is exactly what we need to estimate accurate velocities.

To this end, we propose the first Livox LiDAR assisted 4D perception system with superior and real-time performance. We design our VeloVox to generate 3D proposals at the first stage, then learn per-proposal representation by incorporating a novel Inception architecture to hierarchically enhance the interactions between spatial and temporal point features, and make use of the long-tailing phenomenon rather than avoid. The aggregated features, serving as box query and velocity query, are fed into an attention-based decoder to accomplish precise box refinement and accurate velocity estimation. The

* Equally contributed to the work, which is performed during an internship at Shanghai AI Laboratory. taoma@link.cuhk.edu.hk, zhiwei.zheng@berkeley.edu
[1]Multimedia Laboratory, The Chinese University of Hong Kong [2]Shanghai AI Laboratory [3]University of California, Berkeley.

proposed framework advances promising performance and efficiency, and can be conveniently extended to multi-frame point clouds input with any type of LiDARs.

Additionally, current widely-used datasets [11], [13], [14] for studying autonomous driving only contain point cloud data collected by the mechanical spinning LiDARs. To better verify the effectiveness of our proposed method, we model the basic principle of non-repeating scanning patterns and randomly place several object targets with basic dynamics models in the Carla simulation environment. Therefore, the absolutely accurate velocities of objects are obtained.

The contribution of this work is summarized as follows:

1) We propose VeloVox framework which effectively takes advantage of non-repeating scanning patterns for 4D point-cloud feature learning, leading to improved performance of 3D object detection and velocity estimation with high efficiency.
2) We introduce a 1D Inception network based spatial-temporal interaction module to make effective use of the temporal information in a local-to-global manner.
3) A synthesized dataset is simulated under the basic principle of non-repeating scanning patterns in Carla, which helps to verify the estimated velocity accurately.

II. Related Work

A. 3D Object Detection

Current 3D object detectors usually extract the point cloud feature in grid-based and point-based strategies. The point clouds are transformed into 3D voxels [2], [3], [19], pillars [1], [20], and bird-eye view maps [21], [22] representation by different grid-split designs. Point-based methods [4], [23] often employ PointNet [24], [25] as a base feature extractor. The hybrid strategy [5], [26], [27] is also utilized to leverage both advantages. In addition, the attention mechanism has shown great potential to extract point cloud features. VoTR [28] introduced dilated attention operating on the non-empty voxels to accelerate the convolutional parts. TransFusion [29] initializes the object queries with the proposals' BEV feature. CenterFormer [30] proposes to use the cross-attention transformer to fuse features from multiple different frames. Attention-based detection heads are also utilized to transform object queries into independent 3D bounding boxes [31], [32]. In this paper, we propose two independent attention-based modules to adaptively aggregate the spatial-temporal features for specitic tasks, which are also used to initialize the object queries.

B. Velocity Estimation

Velocity estimation plays a pivotal role in perception systems and following downstream tasks like tracking and planning, particularly in the realm of robotics and autonomous vehicles. The optical flow based methods [33], [34] calculate the motion between two consecutive image frames in a match-and-compare manner. Kalman filtering [35] based methods estimate the velocity based on a series of measurements observed over a recursive time period. NNAKF [36] incorporates an RNN into Kalman filtering, but it assumes a constant velocity, only estimating the direction of the velocity. Most following methods [19], [27], [30], [37] then take as input the concatenation of two or more frames of point clouds, and add one velocity head to tackle the velocity estimation problem. However, these methods only consider the time offset as another attribute of points, which is usually processed with PointNet-like networks to extract features together with the spatial coordinates. In this work, we propose a temporal feature extraction module that incorporates the 1D Inception network, to formulate a spatial feature aggregation along the temporal dimension, leading to accurate velocity estimation performance.

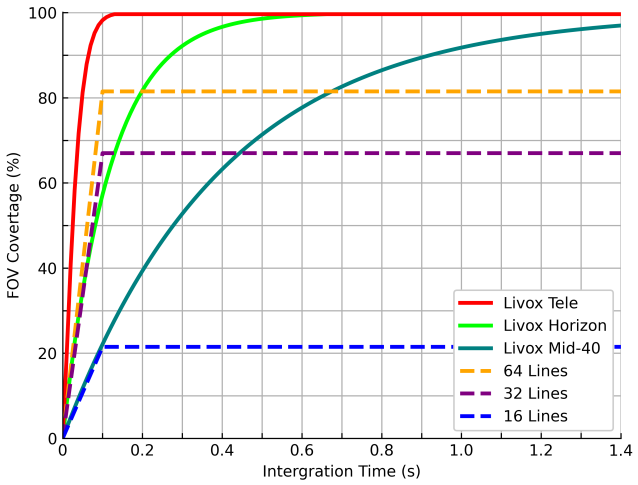

Fig. 2: Comparison of point cloud density between Livox and traditional LiDARs as a function of integration time.

III. Methodology

In this paper, we propose VeloVox, which is a two-stage 4D detection framework aiming at accurate object detection and velocity estimation from point clouds of Livox LiDAR, as illustrated in Fig 3. In the following sections, we will introduce the characteristics of 4D point cloud, the problem statement, and the specific design of each module.

A. Preliminary

We denote $\{p_i = (x, y, z, r, t) \mid i = 1, 2, ..., n\}$ (n points) as one frame of point clouds collected by the Livox LiDAR. The first four characteristics represent the 3D spatial coordinates (x, y, z) and reflectance intensity r, which are the same as traditional mechanical spinning LiDARs. The final item t is a unique temporal characteristic, providing the local timestamp starting from p_0 to the current point p_i, making itself 4-dimensional (*4D point cloud*). Note that the timestamp of the last point p_n is the timestamp of the whole frame.

We aim to simultaneously predict the location (c_x, c_y, c_z), dimension (l, w, h), orientation θ, and motion velocity v of potential objects based on one frame of 4D point cloud. Specifically, the whole VeloVox framework is composed of three parts as shown in Fig. 3, i.e., an RPN backbone for proposal generation, a 1D Inception based spatial-temporal aggregator for proposal feature refinement, and two decoders for proposal refinement and velocity estimation respectively.

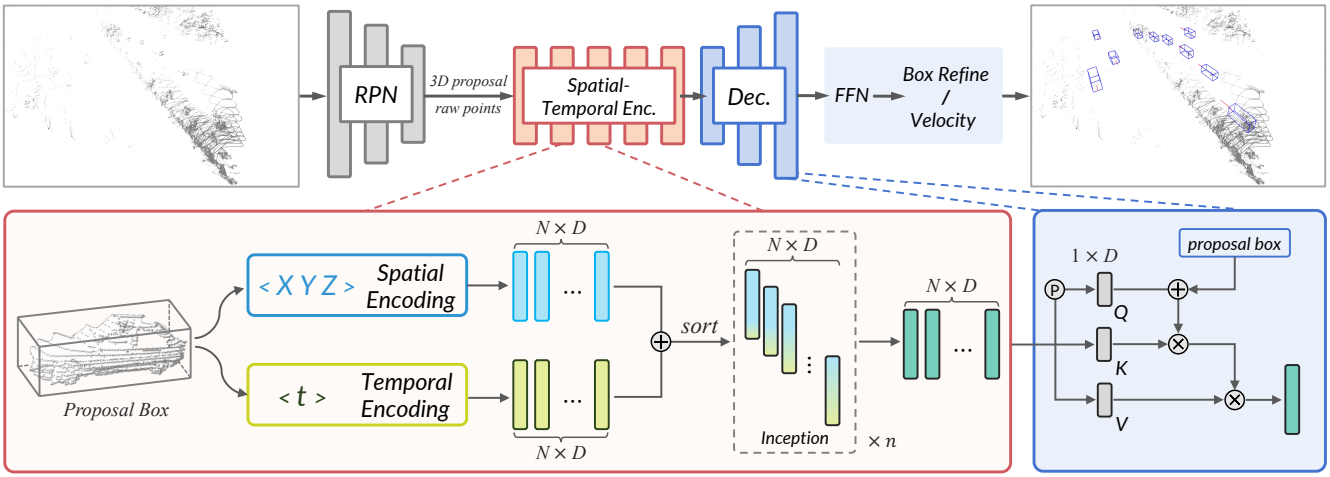

Fig. 3: The framework of our proposed VeloVox. The RPN takes as input the 4D point clouds and generates 3D proposals. Then we take the raw points to encode the spatial (x, y, z) and temporal (t) information. Furthermore, several 1D Inception modules are utilized to enhance the spatial-temporal aggregation along time dimension. Finally, we use attention-based decoders to adaptively output the box refinement and estimated velocity.

B. RPN for Proposal Generation

To better verify the special non-repetitive scanning pattern of the 4D point cloud, we adopt the 3D voxel Center-Point [19] as our default RPN for high efficiency and accuracy. RPN takes as input the points, divided into voxel representations, and generates 3D proposals $\hat{b} = (c_x, c_y, c_z, l, w, h, \theta)$, consisting of center coordinates, dimensions, and orientations. Note that any high-quality RPN should be readily replaceable in our framework and amenable to training via an end-to-end manner.

C. Two-stage Refinement and Velocity Estimation

Given the fact that non-repetitive scanning point clouds naturally carry temporal information, we extract the original points cropped by proposals rather than the voxel features of RPN backbone or the hand-crafted designed grid-pooling features utilized by most SOTA two-stage detectors. The spatial difference between a static and a dynamic car is shown in Fig. 4. The long-tailing phenomenon of dynamic objects makes accurate velocity estimation possible: the number of these tails and the spatial intervals between any two neighboring tails directly reflect the speed of the objects. Based on these observations, we propose a novel spatial-temporal interaction module to make use of the long-tailing phenomenon, and accomplish precise box refinement and accurate velocity estimation, respectively.

1) Spatial Feature Encoding: To take advantage of one-stage proposals for better refinement, we encode proposal information into object points. Specifically, we first operate the local coordinate transform for object points belonging to each proposal [18], such that the object points are more aligned with semantics across different distances. Then, we use a point-to-surface approach to compute the projection distance between each point and the six surfaces of the proposal. Different from the grid-based extraction [5], we randomly sample N points and utilize PointNet [24], [25]

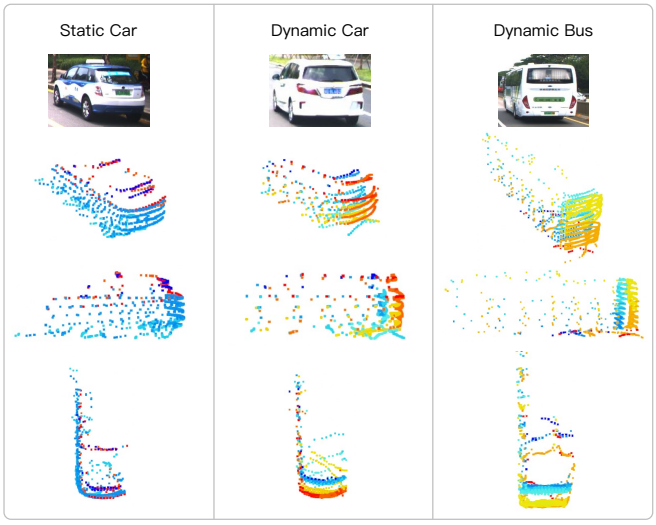

Fig. 4: Comparison of the long-tailing phenomenon for objects in different states of motion. Object points are described in three views and colored based on the timestamps. The greater the velocity is, the more obvious the tailing is.

to learn the point-wise spatial features, because voxelization may erase the tail-like phenomenon if two neighboring tails are so close to being divided into the same voxel. Subsequent multi-layer perceptrons (MLPs) are followed to reduce the dimension, resulting in N 256-dim feature vectors.

2) Temporal Feature Encoding: We first sort all point features according to their timestamps t, leading to regular permutations along the temporal dimension. However, the point-wise movement is still too minimal to distinguish. To better enhance the time span effect, we encode the absolute temporal information as time embeddings through learnable fully connected layers. Similar to the function in Transformers, these time embeddings play a crucial role in expressing temporal effects and revealing sequential patterns.

3) Spatial Temporal Feature Interaction: The spatial features and corresponding time embeddings are first element-

wisely summed together as low-level spatial-temporal features. For the sake of capturing relations among the spatial and motion information, we employ a 1D CNN composed of several 1D Inception modules [38] to process the temporal features. The temporal encoder module converts each local timestamp into a predicted temporal feature embedding. The encoder's basic building block is a 1D Inception module, which consists of 4 1D convolution layers, each with kernel size ranging from 2 to 7 and stride 2 along the time domain, followed by ReLU activation and batch normalization. The Inception module models various ranges of time spans, and enables more accurate information flow from time domain to spatial domain. The number of channels grows from lower layers to higher layers in the stack, in order to reflect the time spans from local to global.

In addition, the skip connection structure is utilized between the local and global spatial-temporal features, for which the same padding is designed to keep the feature dimension not reduced.

4) Decoder: We manage to decode our spatial-temporal point features into the final results, proposal refinement and velocity prediction. Different from the standard detection head, which often aggregates N multiple point features using pooling layers, our decoder attentively utilizes point features according to the following two reasons:

- Surface points are more meaningful for box refinement.
- Points forming long-tailing phenomena are more useful for velocity estimation.

Therefore, it's critically important to vary different levels of point feature combinations. We build two independent cross-attention blocks to enable spatial and temporal interactions respectively. Instead of using a randomly initialized query adopted in previous methods, we directly pool the spatial-temporal fused feature after several MLPs as the learnable query to better respond to the requirements of our prediction tasks. We also encode the original center coordinates and the orientation of proposals as external positional encoding, which is then element-wisely summed into the query. While the former would provide the sparsity information of sampled points, the latter is useful for the direction of velocity. Finally, a feed-forward network (FFN) is leveraged after each block to output the proposal refinement or the velocity estimation.

D. Loss Function

The proposed VeloVox framework is trained in an end-to-end manner, with the region proposal loss L_{rpn} for the 3D proposal generation stage, and the proposal refinement loss L_{ref}, velocity estimation loss L_{velo} for refining stage.

For the first stage, we adopt the same region proposal loss L_{rpn} like SECOND [3] as following:

$$L_{\text{rpn}} = L_{\text{cls}} + \alpha \sum_{r \in \{x,y,z,l,w,h,\theta\}} L_{\text{smooth-L1}}(\hat{r}, r) \tag{1}$$

where L_{cls} is the anchor classification loss calculated with fast focal loss version [39], $L_{\text{smooth-L1}}$ is the residual value regression loss with smooth L1 convergence strategy, and α is the hyper parameter to balance these two losses.

For proposal refining and velocity estimation, we randomly sample 128 proposals with $1:2$ ratio for positive and negative proposals. The loss for the second stage L_{rcnn} is:

$$L_{\text{rcnn}} = \sum_{\Delta r \in \{\Delta x,\Delta y,\Delta z,\Delta l,\Delta w,\Delta h,\Delta\theta\}} L_{\text{smooth-L1}}(\hat{\Delta r}, \Delta r) + \beta L_{\text{cls}} + \gamma L_{\text{velo}} \tag{2}$$

The velocity loss L_{velo} is calculated with L2 loss and we use smooth-L1 loss for box refinement with the predicted residual $\hat{\Delta r}$ and the regression target Δr. β and γ are utilized to balance the weights. During the training, we directly add up the L_{rpn} and L_{rcnn} as our final loss.

IV. Experiments

A. Dataset

We conduct experiments on both the in-house collected real dataset and synthesized dataset. We follow the KITTI [11] official evaluation protocol to report the metric of mAP for object detection, and mean absolute error (MAE) for velocity estimation.

1) In-house Dataset: The in-house dataset contains total 70 scenes, which are collected by 1 Livox LiDAR covering a 60-degree horizontal field of view, and annotated with 3D annotations for *Car* category. Each scene contains 150 consecutive frames at 10Hz. We split them into 55 scenes for training and 15 scenes for validation in our experiments.

2) Synthesized Dataset: Given the high cost of data collection, we build the model of non-repetitive scanning patterns in Carla simulation environment for point cloud generation. Meanwhile, we randomly place several object targets with basic traffic motion models to obtain absolutely accurate velocity values. The motion distortion and local timestamp of the point cloud are also configured through scanning frequency, following the official notes of Livox LiDAR. Note that the reflectance intensity is deprecated in our synthesized data related experiments. The frames of point clouds are 3910 in total, one of which is shown in Fig. 5.

B. Implementation Details

On both datasets, the point cloud range is set as a rectangular region, with $X \in [0, 70.4]$m, $Y \in [-40, 40]$m and $Z \in [-3, 2.5]$m. The voxel size is set to [0.05, 0.05, 0.1]m along each axis for voxel-based method, and $[0.2, 0.2, 5.5]$m for pillar-based method. All the points are processed by eliminating ego motion.

We follow SECOND [3] to implement the data augmentation for the training process. We randomly place several ground-truth objects and corresponding points into the current scene. The height of the target location is constrained by a RANSAC-based [40] ground plane equation. Random

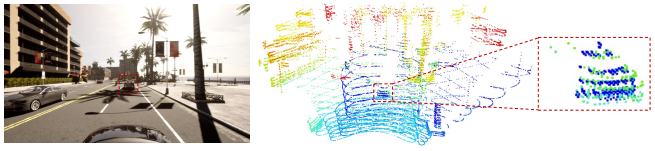

Fig. 5: The visualization of our synthesized data.

Model	Frame	Pred Vel	3D Box ↑			BEV Box ↑			Velocity MAE (m/s) ↓				
			AP70	AP50	AP30	**AP70**	AP50	AP30	≥12	≥64	≥512	≥1024	Avg.
CenterPoint	1		63.64	82.08	88.55	77.13	85.34	89.43	-	-	-	-	-
+ VeloVox	1		**66.13**	**83.40**	**88.99**	**78.75**	**86.32**	**89.84**	-	-	-	-	-
CenterPoint	1	✓	62.69	81.67	88.09	77.21	85.03	89.01	1.62	1.49	1.08	1.39	1.46
+ VeloVox	1	✓	**63.53**	**82.06**	**88.40**	**77.40**	**85.14**	**89.28**	**1.53**	**1.13**	**0.84**	**0.95**	**1.15**
CenterPoint	2	✓	61.04	82.02	88.87	76.43	85.61	89.98	0.78	0.58	0.62	0.52	0.66
+ VeloVox	2	✓	**62.22**	**82.26**	**89.31**	**76.63**	**85.94**	**90.10**	**0.74**	**0.54**	**0.51**	**0.49**	**0.61**
PointPillars	1		62.16	**84.40**	**90.85**	79.41	**88.06**	**91.63**	-	-	-	-	-
+ VeloVox	1		**65.24**	83.97	90.25	**79.53**	87.07	91.01	-	-	-	-	-
PointPillars	1	✓	61.25	84.13	**91.05**	79.36	**88.27**	**91.97**	1.76	1.76	1.59	1.92	1.76
+ VeloVox	1	✓	**65.05**	**84.14**	90.51	**79.38**	87.73	91.32	**1.57**	**1.25**	**1.03**	**0.93**	**1.28**
PointPillars	2	✓	58.63	83.25	**90.40**	77.60	87.19	**91.30**	0.89	0.70	0.61	0.64	0.76
+ VeloVox	2	✓	**63.22**	**84.12**	90.32	**78.90**	**87.48**	91.20	**0.78**	**0.57**	**0.50**	**0.47**	**0.64**
SECOND	1		64.32	83.94	**90.54**	79.26	87.59	**91.57**	-	-	-	-	-
+ CT3D	1		65.67	84.16	90.44	78.55	87.60	91.31	-	-	-	-	-
+ VeloVox	1		**66.10**	**84.28**	90.40	**79.72**	**87.72**	91.20	-	-	-	-	-
SECOND	1	✓	63.12	83.93	90.18	78.75	87.41	91.06	1.76	1.53	1.47	1.28	1.52
+ CT3D	1	✓	63.91	**84.07**	**90.65**	78.30	**87.61**	**91.51**	1.71	1.32	1.01	1.03	1.31
+ VeloVox	1	✓	**65.53**	83.94	90.20	**79.20**	87.31	91.07	**1.57**	**1.19**	**0.83**	**0.82**	**1.16**
SECOND	2	✓	62.56	84.93	**91.85**	78.68	**88.78**	**92.83**	0.80	0.59	0.54	0.49	0.66
+ CT3D	2	✓	63.92	85.06	91.68	78.80	88.69	92.57	0.78	0.57	0.53	0.55	0.64
+ VeloVox	2	✓	**64.60**	**85.21**	91.33	**78.81**	88.66	92.03	**0.77**	**0.56**	**0.50**	**0.47**	**0.63**

TABLE I: Performance comparison of 3D detection and velocity estimation on the *Car* class with our proposed VeloVox on different backbones. We also provide results for CT3D with SECOND. MAE is listed by the number of object points.

world flipping along X-axis, random scaling with a ratio of $[0.95, 1.05]$, and random rotation uniformly sampled from $[-\pi/8, \pi/8]$ are adopted for the global point cloud frame.

We train our models using Adam optimizer with an initial learning rate of 1e-4, which is modified using the one-cycle learning policy. The decay weight is set to 0.01, and the momentum range is [0.95, 0.85]. The model is trained through 80 epochs with a batch size of 16.

C. Main Results

Extensive experiments are conducted to evaluate the effectiveness of each proposed module. We implement PointPillars [1], SECOND [3] and CenterPoint [19] as our baselines, together with the corresponding multi-frame versions to enable the comparison of velocity estimation. Additionally, the synthesized data is utilized to better verify the performance of velocities. Qualitative results of VeloVox on the in-house validation part are shown in Fig 6.

1) Evaluation of 4D Object Detection: We compare our VeloVox on several previous 3D detectors in Table I with AP of 3D and BEV views, and velocity MAE. For the traditional models, the proposed spatial-temporal encoding and decoder are replaced with an additional velocity head. Our proposed VeloVox could improve both the detection AP and velocity accuracy with considerable gains. Note that, the detection AP of 2-frame CenterPoint is slightly lower than 1-frame version, which is opposed to the conclusion drawn from mechanical spinning LiDARs. We infer that the long-tailing problem will be more serious with Livox LiDAR. Besides, with our VeloVox, the estimated velocities are more accurate than the corresponding baselines.

2) Evaluation of Two-stage: We conduct experiments with different combinations used in two-stage models for velocity estimation. Models are trained and validated with the input of ground-truth boxes.

We provide the experiment results in Table II. We take the 1st row, which merely adopts the network proposed in PointNet [24], as the baseline. It only takes spatial information into consideration, ignoring temporal information t. If we do not sort the feature by temporal information, the Inception modules make few contributions (2nd and 3rd rows). Through temporal encoding of absolute timestamps, it significantly exceeds the baselines, as shown by 4th row. With the spatial-temporal aggregated pooling query or sampling strategy based on the time window, the performance can be further improved. As shown by the 7th row, when we combine all these sub-modules together, the performance improves a lot, enhancing the performance on different groups by 34.80%, 43.01%, 53.26%, 61.79%, and 40.97% respectively.

3) Evaluation on Synthesized Dataset: In order to accurately verify the correctness of estimated velocities, we ran the experiment on the synthesized dataset. The MAE of estimation results is 0.8298, showing the effect of our method, and the performance is better than that on the real dataset, which may be caused by the noise of human annotations on real data.

4) Attention Mechanism: To evaluate whether the two branches of the decoder attend to different points, we visualize the attention maps of the cross-attention layer. As shown in Fig. 7, the blue and red points represent the results of top-30 weights for box refinement and velocity estimation, respectively. Most of the blue points are distributed on the corners and surfaces, which contribute much to the precise geometry prediction. The red points are concentrated around the rear parts, which are exactly the long-tailing points. It not only proves the effectiveness of our attention-based decoder, but also shows the function of spatial-temporal interactions.

D. Resource Analysis

Real-time speed is a very important indicator if we want to deploy the algorithms to autonomous vehicles or robots. We

Sub-modules							Velocity MAE (m/s)				
PointNet	Sort	Inception	Temporal	Decoder	Max-pool Query	Sample by time	[12, 128]	[128, 512]	[512, 1024]	[1024,)	Avg.
✓							2.27	2.16	1.84	3.01	2.27
✓		✓					2.24	2.12	1.94	3.09	2.26
✓	✓	✓					1.84	1.56	1.17	1.65	1.70
✓	✓	✓	✓				1.58	1.38	1.098	1.06	1.47
✓	✓	✓	✓	✓			1.59	1.32	0.98	1.33	1.45
✓	✓	✓	✓	✓	✓		1.50	1.26	**0.85**	1.20	1.36
✓	✓	✓	✓	✓	✓	✓	**1.48**	**1.23**	0.86	**1.15**	**1.34**

TABLE II: Performance comparison for different combinations of modules. The velocity MAE is reported based on the original number of object points.

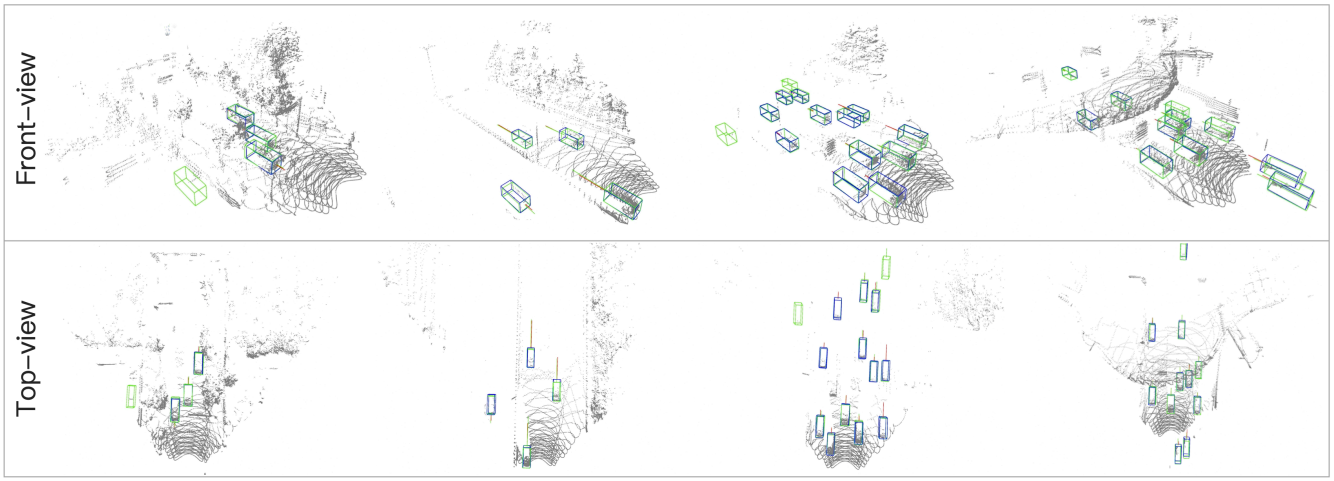

Fig. 6: The qualitative results of our VeloVox. The predicted and ground-truth boxes are colored blue and green respectively. The estimated velocities are colored in red. Please zoom in for better visualization.

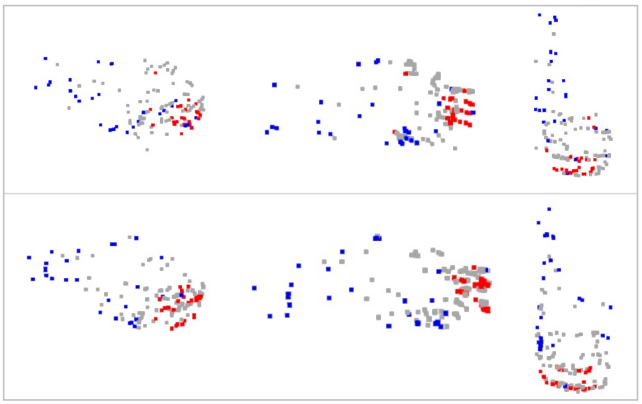

Fig. 7: Attention maps generated by the cross-attention layer. We visualize the points of top-30 weights for box refinement (blue) and velocity estimation (red).

measure the runtime and the required memory with an Intel i7 CPU and an NVIDIA RTX 3090 GPU. The batch size is set to 1. Our VeloVox achieves around 2949M memory and 68ms per frame, which is lower than the 10Hz frame rate.

E. Sample Points

We separately sample 64, 128, and 256 points to evaluate the impact of points number. The velocity MAE metric is reported on the in-house val set, where objects are divided into 4 groups by original points number. Table III shows that as we sample more points, the performance on the objects with more points is also improved. We keep the stride of 1D Inception modules unchanged, the design is acceptable for objects with up to 512 points after sampling, which also attributes to our attention-based decoder.

numbers of points	Velocity MAE (m/s)				
	[12, 128]	[128, 512]	[512, 1024]	[1024,)	Avg.
64	1.48	1.31	0.87	1.21	1.36
128	1.48	**1.23**	**0.86**	1.15	1.34
256	**1.47**	**1.23**	0.88	**1.07**	**1.33**

TABLE III: Performance comparison with different numbers of sampled points.

V. Conclusion

In this paper, we proposed a novel 4D object detection method with the single-frame point cloud of Livox LiDAR. With the 3D proposals generated by different RPN backbones, we separately encode the spatial and temporal features. Then, a series of 1D Inception modules is leveraged to enhance spatial-temporal feature interaction, making use of the long-tailing phenomenon of 4D point clouds. The decoder consists of two attention-based branches to adaptively predict the box refinement and velocities. Quantitative results and ablation studies on both in-house and synthesized datasets demonstrate the effectiveness and outstanding performance of our VeloVox.

Acknowledgement

This project is funded in part by Shanghai Artificial Intelligence Laboratory, National Key R&D Program of China Project 2022ZD0160104 and the Science and Technology Commission of Shanghai Municipality 22DZ1100102, and in part by General Research Fund of Hong Kong RGC Project 14204021.

REFERENCES

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 5

[2] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[3] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," in *Sensors*, 2018. 1, 2, 4, 5

[4] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[5] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3

[6] M. Aygun, A. Osep, M. Weber, M. Maximov, C. Stachniss, J. Behley, and L. Leal-Taixé, "4d panoptic lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5527–5537. 1

[7] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *Proceedings of the IEEE Conference Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 359–10 366. 1

[8] Q. Wang, Y. Chen, Z. Pang, N. Wang, and Z. Zhang, "Immortal tracker: Tracklet never dies," *arxiv preprint*, 2021. [Online]. Available: https://arxiv.org/abs/2111.13672 1

[9] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," *arxiv preprint*, 2021. [Online]. Available: https://arxiv.org/abs/2111.09621 1

[10] J. V. Hurtado, R. Mohan, W. Burgard, and A. Valada, "Mopt: Multi-object panoptic tracking," *arXiv preprint arXiv:2004.08189*, 2020. 1

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013. 1, 2, 4

[12] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307. 1

[13] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454. 1, 2

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631. 1, 2

[15] Z. Yang, Y. Zhou, Z. Chen, and J. Ngiam, "3d-man: 3d multi-frame attention network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1

[16] X. Chen, S. Shi, B. Zhu, K. C. Cheung, H. Xu, and H. Li, "Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection," in *Proceedings of the European conference on computer vision (ECCV)*, September 2022. 1

[17] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov, "Offboard 3d object detection from point cloud sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1

[18] T. Ma, X. Yang, H. Zhou, X. Li, B. Shi, J. Liu, Y. Yang, Z. Liu, L. He, Y. Qiao, Y. Li, and H. Li, "Detzero: Rethinking offboard 3d object detection with long-term sequential point clouds," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2023. 1, 3

[19] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2, 3, 5

[20] Y. Wang, A. Fathi, A. Kundu, D. A. Ross, C. Pantofaru, T. A. Funkhouser, and J. M. Solomon, "Pillar-based object detection for autonomous driving," in *The European Conference on Computer Vision (ECCV)*, 2020. 2

[21] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting hd maps for 3d object detection," in *Conference on Robot Learning*. PMLR, 2018, pp. 146–155. 2

[22] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021. 2

[23] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017. 2, 3, 5

[25] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:2212.07289*, 2022. 2, 3

[26] J. Noh, S. Lee, and B. Ham, "Hvpr: Hybrid voxel-point representation for single-stage 3d object detection," *arXiv preprint arXiv:2104.00902*, 2021. 2

[27] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[28] J. Mao, Y. Xue, M. Niu *et al.*, "Voxel transformer for 3d object detection," *ICCV*, 2021. 2

[29] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[30] Z. Zhou, X. Zhao, Y. Wang, P. Wang, and H. Foroosh, "Centerformer: Center-based transformer for 3d object detection," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Springer, 2022, pp. 496–513. 2

[31] I. Misra, R. Girdhar, and A. Joulin, "An End-to-End Transformer Model for 3D Object Detection," in *ICCV*, 2021. 2

[32] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, "Improving 3d object detection with channel-wise transformer," in *ICCV*, 2021, pp. 2743–2752. 2

[33] L. Shindler, M. Moroni, and A. Cenedese, "Using optical flow equation for particle detection and velocity prediction in particle tracking," *Applied Mathematics and Computation*, vol. 218, no. 17, pp. 8684–8694, 2012. 2

[34] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 433–466, 1995. 2

[35] R. Kalman, "A new approach to linear filtering and prediction problems," in *Journal of Basic Engineering*, 1960. 2

[36] S. Jouaber, S. Bonnabel, S. Velasco-Forero, and M. Pilté, "Nnakf: A neural network adapted kalman filter for target tracking," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4075–4079. 2

[37] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932. 2

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. 4

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988. 4

[40] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved ransac for 3d point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, 2017. 4