Future Does Matter: Boosting 3D Object Detection with Temporal Motion Estimation in Point Cloud Sequences

Rui Yu[1]
y80220166@mail.ecust.edu.cn
Runkai Zhao[2]
rzha9419@uni.sydney.edu.au
Cong Nie[3]
2132586@tongji.edu.cn
Heng Wang[2]
hwan9147@uni.sydney.edu.au
HuaiCheng Yan[1]
hcyan@ecust.edu.cn
Meng Wang[1]
mengwagn@ecust.edu.cn

[1] East China University of Science and Technology, Shanghai, China
[2] University of Sydney, Sydney, Australia
[3] Tongji University,Shanghai, China

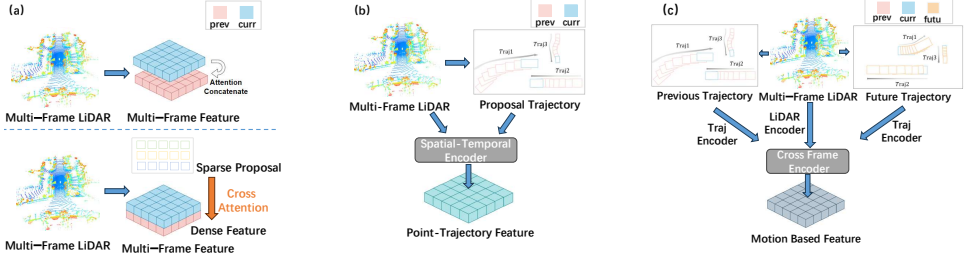

Figure 1: Different from the global bird's eye view (BEV) Neighbor Feature Fusion Method (a) and Trajectory-based Method (b) which do not count for the role of the future states, we propose a novel LiDAR 3D object detection framework that utilizes motion forecasting to guide the temporal fusion learning across past and future frames as shown in (c).

Abstract

Accurate and robust LiDAR 3D object detection is essential for comprehensive scene understanding in autonomous driving. Despite its importance, LiDAR detection performance is limited by inherent constraints of point cloud data, particularly under conditions of extended distances and occlusions. Recently, temporal aggregation has been proven to significantly enhance detection accuracy by fusing multi-frame viewpoint information

and enriching the spatial representation of objects. In this work, we introduce a novel LiDAR 3D object detection framework, namely ***LiSTM***, to facilitate spatial-temporal feature learning with cross-frame motion forecasting information. We aim to improve the spatial-temporal interpretation capabilities of the LiDAR detector by incorporating a dynamic prior, generated from a non-learnable motion estimation model. Specifically, Motion-Guided Feature Aggregation (MGFA) is proposed to utilize the object trajectory from previous and future motion states to model spatial-temporal correlations into gaussian heatmap over a driving sequence. This motion-based heatmap then guides the temporal feature fusion, enriching the proposed object features. Moreover, we design a Dual Correlation Weighting Module (DCWM) that effectively facilitates the interaction between past and prospective frames through scene- and channel-wise feature abstraction. In the end, a cascade cross-attention-based decoder is employed to refine the 3D prediction. We have conducted experiments on the Waymo and nuScenes datasets to demonstrate that the proposed framework achieves superior 3D detection performance with effective spatial-temporal feature learning. https://github.com/YuRui-Learning/LiSTM

1 Introduction

3D LiDAR object detector [11, 34, 37] plays an important role in autonomous driving, it identifies object information within a 3D road scene represented by an unstructured point cloud. Although discrete LiDAR points reflect accurate spatial positioning of surrounding driving scenes, they are insufficient to comprehensively describe traffic objects due to data sparsity, particularly at far distances. Moreover, the LiDAR sensor captures partial view information of a scene from a single-frame perspective, leading to incomplete information collection of the visible objects. These inherent limitations of LiADR result in inconsistent point distribution for the same object across a driving sequence. Hence, a dynamic object may be represented with varying densities of point clouds in different frames, which introduces ambiguity in accurately determining the true shape for a 3D detector.

To eliminate the inconsistency, the increasing works [2, 33, 38] attempt to detect 3D objects by utilizing multiple frames of point clouds. The LiDAR sensor records driving scenarios as the vehicle moves, delineating objects across multiple perspectives in sequence. This adds valuable modal information, enriching object representation. A straightforward method to implement this idea is to fuse the neighboring frame features, using the insight of historical frames to enhance the semantic representation of the current scene. Referring to the application of transformer in computer vision, the cross-attention mechanism bridges the previous and current point features either densely or sparsely, as depicted in Figure 1(a).

Direct integration of features for historical frames enhances the detection performance, but this method struggles to handle fast-moving objects. To solve this issue, trajectory-based methods [3, 8] are designed to aggregate extensive temporal contexts of the object flows and utilize multi-frame proposals to comprehend the spatial information among the driving scenes. As shown in Figure 1(b), this method enhances the representation of the object by incorporating multi-view complementary information from the corresponding trajectory. However, this input-level manipulation is resource-intensive, limiting detection efficiency.

To boost temporal object detection, we propose a novel **Li**DAR 3D object detection with enhancing **S**patial-**T**emporal feature fusion through **M**otion estimation, namely ***LiSTM***. We bolster spatial-temporal feature fusion by integrating a Kalman filter module [10] as prior kinetic information and focus on effectively integrating both ego and object motion states. Unlike previous approaches [3, 8] that directly encode proposal trajectories with point clouds,

we uncover an implicit feature representation for both trajectories and point clouds within the BEV space using the motion-based heatmap generator. This enables direct feature-level fusion, eliminating the need for reliance on the PointNet [18] backbone. To have a stronger dynamic prior for each frame, we design the **M**otion-**G**uided **F**eature **A**ggregation (MGFA) mechanism to combine the heatmap generated by trajectory prediction for guiding the reconstruction of LiDAR features. Ultimately, with the integration of the **D**ual **C**orrelation **W**eighting **M**odule (DCWM) and Motion Transformer, we enhance feature characterization across frames, thereby enriching the semantic and geometric representations.

The main contributions of this paper can be summarized as follows:

- We propose a novel LiDAR object detector considering future motion estimation of objects and point clouds to enhance the effectiveness of the spatial-temporal fusion.
- We design a Motion-Guided Feature Aggregation (MGFA) mechanism to enhance object geometric representations of motions, and the Dual Correlation Weighting Module (DCWM) to characterize the spatial relationship of features across sequences.
- We conduct experiments on the nuScenes and Waymo datasets to validate our proposed framework, which outperforms CenterPoint by 8% on the Waymo dataset.

2 Related Work

BEV 3D Object Detection. The bird's-eye view (BEV) is a widely used feature representation in the field of autonomous driving which is derived from LiDAR's ability to perceive objects from a circular viewpoint. Thanks to the PointNet series [18], point-based methods [20] have become extensively employed to extract geometric features directly from point clouds. Voxel-based methods [4, 35, 37] and Pillar-based methods [11, 19] are mainly applied in environmental perception by converting point cloud to BEV feature. Meanwhile, Camera-based detectors [14, 17] learn pixel-wise categorical depth distributions to lift 2D images of different views into BEV space. Additionally, Li *et al*. [15] proposes a spatiotemporal transformer and focus on feature fusion in the spatial-temporal 4D working space.

Keypoint Detection. Anchor-based methods [16, 26] often result in redundant bounding boxes, requiring the use of Non-Maximum Suppression. Law and Deng [12] produce two corner pairs to detect, while Zhou *et al*. [36] uses keypoint estimation with a normal distribution to locate center points, which use the central region to regress other properties. Therefore, CenterPoint [34] follows the struct of CenterNet [36] and employs an object detector in BEV space. Zhou *et al*. [38] utilizes the initial query embedding to facilitate learning of the transformer and uses cross attention to efficiently aggregate neighboring features.

Temporal Fusion Methodology. Temporal Fusion plays a critical role in autonomous driving, allowing models to gain a deeper understanding of contextual geometric information. Zhou *et al*. [38] performs multi-frame features fusion by utilizing spatial-aware attention, while RNN-based models [2, 33] employ LSTM and GRU to fuse previous state features with the current feature. BEVFormer [15] designs a temporal deformable attention to fuse previous features for enhanced performance. Meanwhile, Wang *et al*. [28] develops an object-centric temporal mechanism and a motion-aware layer normalization to model the movement of the objects. 3D-MAN [32] utilizes a multi-frame alignment and aggregation module to learn temporal attention for detection from multiple frames. motion-based models [3, 8, 9] design point-trajectory transformer with long short-term memory for efficient

temporal 3D object detection. Li *et al.* [13] uses motion forecasting outputs as a type of virtual lightweight sensor modality. Hence, we propose a more powerful and efficient spatial-temporal fusion model under BEV using CenterPoint [34] as the baseline.

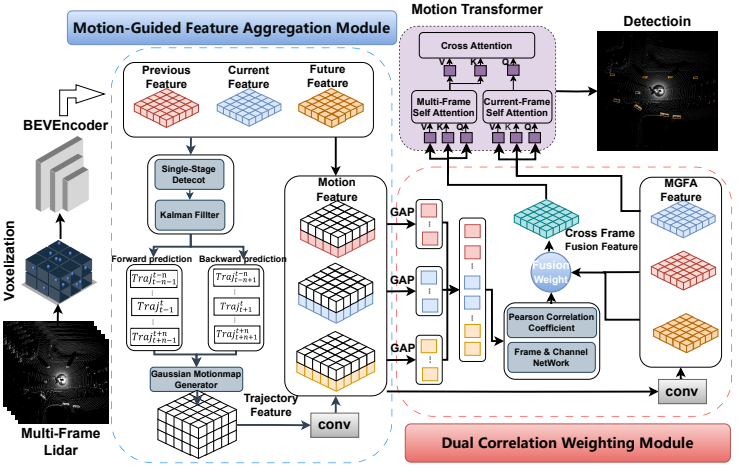

Figure 2: Overview of our proposed framework ***LiSTM***. It processes multi-frame point clouds by performing voxelization before feeding them into the LiDAR BEV encoder. The first module employs a single-stage detector combined with tracking prediction to produce trajectories and then enhances the spatial representation with a Motion-Guided Feature Aggregation Module. The second module is used for cross-frame feature extraction by the proposed Dual Correlation Weighting Module and Motion Transformer.

3 Approach

As depicted in Figure 2, to incorporate the motion prior, we focus in Section 3.1 on the generation of the motion feature and the Motion-Guided Feature Aggregation (MGFA) mechanism. Then, the Dual Correlation Weighting Module (DCWM) and the Motion Transformer will be presented in Sections 3.2 and 3.3 to describe the cross-frame fusion strategy.

3.1 Motion-Guided Feature Aggregation

Unlike the early fusion methods [9, 8], we utilize motion-based heatmap representing temporal streams to normalize features of objects for deep fusion. To predict object positions in future scenes, we use a kinematic model of ego-motion to derive the transformation matrix from time t to $t+n$, based on prior motion data and ego-pose observations. The transformation matrix is then used to transfer the point cloud in the current scene to a future coordinate, but it only applies to static objects and obtains a coarse-grained prediction. However, whether the points are predictions or observations are processed through voxelization and encoder to produce features $F_{multi} = \{F_{t-n}, ..., F_{t+n}\}$. Then as implemented in CenterPoint [34], we can get multi-frame proposals, which are temporal independence and geometric correlation.

Motion Model. After acquiring multiple consecutive frames of object proposals, we can use a Kalman filtering [10] to estimate the motion state of each object across the frames. We

define a ten-dimension state space $(x,y,z,\theta,l,w,h,\dot{x},\dot{y},\dot{z})$, where $B=(x,y,z)$ is the center of a 3d bounding box, $P_{dim}=(l,w,h)$ is the object size, θ is the orientation under BEV and $V=(\dot{x},\dot{y},\dot{z})$ are the respective velocities in the 3D space learned by a Kalman filter for constant velocity motion with a linear observation model.

Trajectory Prediction. With the Kalman filter modeling multiple targets over a driving sequence, we can obtain information about the velocity prediction V^t of each proposal at every moment. For the forward trajectory prediction, we utilize the bounding box observation B_{t-1} at $t-1$, along with the velocity prediction V_{t-1}, to update the $B't$ for frame t. Similarly, for the reverse trajectory prediction, we employ the bounding box observation B_{t+1} at $t+1$ and the updated velocity prediction V_{t+1} to reverse-predict the predicted the B''_t:

$$B'_t = B_{t-1} + V^{t-1} \cdot \Delta t, \tag{1}$$

$$B''_t = B_{t+1} - V^{t+1} \cdot \Delta t. \tag{2}$$

Motion-based Heatmap Generator. After acquiring the forward and backward trajectory predictions B'_t and B''_t, we transfer these trajectories into motion feature F_{motion} using gaussian distribution. As is known, gaussian distribution is determined as:

$$\mu_k^x = cx_k, \quad \mu_k^y = cy_k, \tag{3}$$

where μ_k represents the location of the proposal under BEV, and σ_k is the hyperparameter of the category associated with the category of the k^{th} object.

For the normal representations $N^t_{t-1}(\mu_k,\sigma_k^2)$ and $N^t_{t+1}(\mu_k,\sigma_k^2)$ of each frame proposal generated by bidirectional trajectory prediction, We respectively use the σ_k to control the probability of the distribution and the μ_k to represent the center of the distribution. Given the proposals from neighboring frames, we can consolidate all distributions into the BEV representation F_{motion}, which enhances the understanding of agent objects by providing additional motion modality insights. This can be very effective in solving fast-moving objects and supplying a prior for occlusion situations.

Motion Guided Feature Aggregation Module. The designed MGFA module utilizes the information from previous and future motion states to interact with dense BEV features to model spatial-temporal correlations. By incorporating F_{motion}, we can enrich positional semantic information and integrate motion characterization into the model's understanding. As mentioned, the motion feature includes bidirectional projections for the target frame. Therefore, the specific motion features are denoted as follows, with $p2c$ and $f2c$ representing past and future predictions of the current frame, respectively:

$$F^{p2c}_{motion} = \{N^{t-n}_{t-n-1}, ...N^t_{t-1}, ..., N^{t+n}_{t+n-1}\}, \tag{4}$$

$$F^{f2c}_{motion} = \{N^{t-n}_{t-n+1}, ...N^t_{t+1}, ..., N^{t+n}_{t+n+1}\}. \tag{5}$$

Based on the given feature F_{motion}, it is first expanded along the channel dimension and then processed by a shared $Conv$ to encode the geometric information of the target center. Specifically, $Conv$ denotes channel expansion followed by dimensionality reduction within the channel dimension:

$$F^{p2c/f2c}_{center} = Conv(repeat(F^{p2c/f2c}_{motion})). \tag{6}$$

After obtaining the center distribution feature $F^{p2c/f2c}_{center}$, we follow the method illustrated in Figure 3 to perform the feature fusion using a shared convolutional network. In the aggregation of forward prediction, we merge the forward distribution feature F^{p2c}_{center} of the target

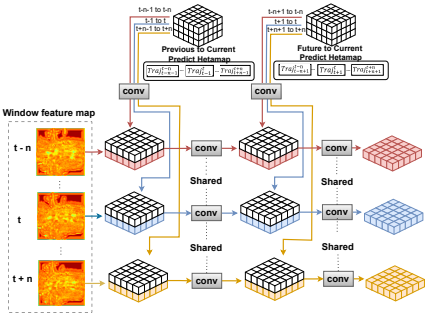

Figure 3: Motion Guided Feature Aggregation.

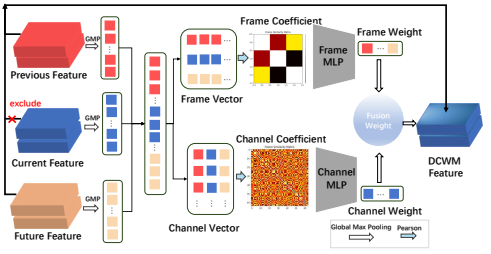

Figure 4: Dual Correlation Weighting Module.

frame from the previous frame with the BEV feature by using a convolutional network. It mainly convolves the channel dimension to realize the fusion of heterogeneous features:

$$F_{MGFA} = Conv(F_{multi}, F^{p2c}_{center}) = \{Conv(F_{t-n}, N^{t-n}_{t-n-1}), ..., Conv(F_{t+n}, N^{t-n}_{t+n-1})\}. \quad (7)$$

Similarly, in reverse trajectory prediction, the center distribution feature F^{f2c}_{center} is sequentially concatenated and convolved with the BEV feature to enhance the dynamic property:

$$F'_{MGFA} = Conv(F_{MGFA}, M^{f2c}_{center}) = \{Conv(F'_{t-n}, N^{t-n}_{t-n+1}), ..., Conv(F'_{t+n}, N^{t-n}_{t+n+1})\}. \quad (8)$$

3.2 Dual Correlation Weighting Module

Unlike the feature concatenation in Figure 1(a), we propose learning a multi-frame fusion weight matrix to capture cross-frame correlations in both channel and temporal dimensions. As shown in Figure 4, global max pooling (GMP) is first applied along the spatial dimensions to obtain a feature vector v_t. Subsequently, vectors from multiple frames are concatenated to form a representation for the scene sequence data, denoted as $V = \{v_{t-n}, ..., v_{t+n}\}$:

$$V = Concat(v_{t-n}, .., v_{t+n}) = Concat(GMP\{F'^{t-n}_{MGFA}\}, .., GMP\{F'^{t+n}_{MGFA}\}), \quad (9)$$

$$M_{d/t} = \frac{conv(V_i^{d/t}, V_j^{d/t})}{\sigma_{V_i^{d/t}} * \sigma_{V_j^{d/t}}}. \quad (10)$$

We then compute the correlation between matrices across each vector(e.g., i and j), where $V^{d/t}$ denotes the process of transforming the sequence along the channel and temporal. After obtaining the correlation matrices M_d and M_t, which represent interlinks within the feature structure and across frames in the temporal domain, respectively, the weight matrix is flattened and passed through a two-layer linear network with ReLU activation:

$$W_{d/t} = Linear(ReLu(Linear(M_{d/t}))). \quad (11)$$

Eventually, we obtain weight vectors $W^{d/t}$ for channels and temporal dimensions, respectively, and generate the weight matrix M_{weight} through their outer product $\otimes$. Then, this weight is multiplied and channel-wise convolution with the MGFA feature F''_{MGFA} (excluding the current frame) to generate the Dual Correlation Weighting feature F_{DCWM} as follows:

$$F_{DCWM} = Conv(F''_{MGFA} \cdot M_{weight}) = Conv(F''_{MGFA} \cdot (W_d \otimes W_t)). \quad (12)$$

Model	Frames	Vehicle (AP/APH)↑		Pedestrian (AP/APH)↑		Cyclist (AP/APH)↑	
		L1	L2	L1	L2	L1	L2
PointPillar [10]	1	66.94 / 66.36	58.96 / 58.43	63.35 / 45.22	55.21 / 39.32	55.06 / 52.55	52.97 / 50.55
VoxelNet [50]	1	68.73 / 67.31	60.11 / 59.97	69.65 / 57.38	60.19 / 53.67	62.31 / 59.85	60.34 / 55.89
PillarNet [35]	1	66.29 / 65.63	59.03 / 58.43	70.35 / 64.24	64.24 / 55.75	65.43 / 63.93	63.53 / 62.08
Second [53]	1	68.95 / 68.33	61.81 / 61.24	65.59 / 54.80	57.85 / 48.16	61.14 / 59.50	56.84 / 55.26
CenterPoint [54]	1	67.87 / 67.27	59.96 / 59.43	69.31 / 62.55	61.17 / 55.06	64.28 / 63.05	61.86 / 60.68
PartA2 [41]	1	65.52 / 64.85	57.32 / 56.63	54.83 / 37.72	46.85 / 32.19	54.29 / 48.75	52.21 / 46.89
PVRCNN [42]	1	71.11 / 70.32	62.60 / 61.88	63.63 / 32.77	54.88 / 28.26	59.49 / 34.14	57.22 / 32.83
VoxelRCNN [8]	1	71.51 / 70.98	63.75 / 63.26	65.95 / 65.99	65.47 / 60.86	70.11 / 68.71	67.98 / 66.63
CenterPoint [54]	4	71.27 / 70.73	63.59 / 63.09	73.91 / 70.45	66.28 / 60.10	63.78 / 62.98	61.59 / 60.82
CenterPoint [54]	16	72.53 / 71.31	64.18 / 64.21	74.05 / 71.17	66.17 / 61.03	64.05/ 64.54	62.31 / 61.77
MPPNet [3]	4	74.24 / 73.55	66.29 / 65.38	76.94 / 72.29	68.63 / 66.16	67.34/ 66.67	65.12 / 64.48
MSF [3]	4	74.37 / 73.97	66.35 / 65.85	**78.16 / 74.91**	**70.27 / 67.21**	67.89/ 67.14	65.58 / 64.89
LiSTM	3	**74.83 / 74.32**	**66.85 / 66.17**	75.89 / 69.72	66.83 / 63.43	**70.84 / 69.75**	**68.23 / 69.12**

Table 1: Quantative comparisons on 20% Sequence Waymo validation set.

Model	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
PointPillar [10]	58.62	45.27	0.3353	0.259	03286	0.2784	0.2002
Second [53]	62.31	50.8	0.3140	0.2554	**0.2785**	0.2587	0.2019
CenterPoint [54]	66.29	58.77	0.2919	0.2566	0.3692	**0.2081**	**0.1837**
VoxelNext [9]	67.09	60.55	0.3023	0.2526	0.3701	0.2087	0.1851
LiSTM	**68.32**	**63.77**	**0.2895**	**0.2479**	0.3182	0.2472	0.1850

Table 2: Quantative comparisons on nuScenes validation set.

3.3 Motion Transformer

With the assistance of the designed modules MGFA and DCWM, the features are enhanced to include details about both ego-motion and object-motion. The attention mechanism [77] is then employed using a transformer decoder to focus on feature learning within the spatial-temporal 4D space. First, the features are processed through self-attention as follows:

$$Q_{C/M} = MultiHeadAttn(Q(F_{C/M}+PE), K(F_{C/M}+PE), V(F_{C/M})), \tag{13}$$

where $F_{C/M}$ represents the current frame feature and DCWM feature, $Q_{C/M}$ denotes the query and PE is the position embedding. After self-attention, we make a cross-attention mechanism with Q_C and Q_M, which guides the training to focus on aggregating more spatial information containing meaningful object details. Then, the cross-attention is shown below:

$$Q'_C = MultiHeadAttn(Q(Q_C+PE), K(Q_M+PE), V(Q_M)). \tag{14}$$

After feature generation and fusion, we get the final target characterization Q'_C. Then we follow the steps of CenterPoint [54] to learn the representation of the different geometric elements in the 3D scene.

4 Experiments

Dataset and Metrics. The Waymo Open dataset [74] is a highly regarded benchmark for automatic driving. It consists of 1150 point cloud sequences, with over 200,000 frames in total. Evaluation of results using mean Average Precision (mAP) and its weighted variant by heading accuracy (mAPH). Results are reported for LEVEL 1 (L1, easy only) and LEVEL 2 (L2, easy and hard) difficulty levels, considering vehicles, pedestrians, and cyclists.

The nuScenes dataset [1] provides diverse annotations for autonomous driving and features challenging evaluation metrics. These include mean Average Precision (mAP) at four

CenterPoint	MotionTransformer	MGFA	DCWM	Veh. L2 APH	Ped. L2 APH	Cyl. L2 APH
✓	×	×	×	59.51	55.22	60.54
✓	✓	×	×	62.49	56.04	63.17
✓	✓	✓	×	64.67	57.56	67.86
✓	✓	✓	✓	**65.88**	**61.10**	**68.8**

Table 3: Ablation studies on Waymo validation set.

center distance thresholds and five true-positive metrics: ATE, ASE, AOE, AVE, and AAE, which measure translation, scale, orientation, velocity, and attribute errors, respectively. Additionally, the nuScenes detection score (NDS) combines mAP with these metrics.

Experimental Settings. In our experimental setup, we follow the default settings of Openpcdet [25] and conduct the experiments using two 24GB Nvidia RTX 3090 GPUs. The validation process utilized the nuScenes and Waymo datasets. We employed the AdamW optimizer with a base learning rate of 3×10^{-3} and applied layer-wise learning rate decay.

Comparison Experiment. We validate the effectiveness of the designed LiSTM on Waymo's validation set (Table 1), using 20% of the sequences for training. Full results are available in Table 8 of the Appendix. LiSTM achieves an impressive improvement of over 8% compared to single-stage models like CenterPoint [34], while also outperforming two-stage models such as PVRCNN [22] and VoxelRCNN [6]. Meanwhile, LiSTM, a multi-frame single-stage model, eliminates the need for region-of-interest extraction, resulting in reduced resource consumption, as illustrated in Table 7. In comparison to multi-frame CenterPoint [34], LiSTM achieves remarkable improvements while utilizing fewer frames. When compared to two-stage models MPPNet [3] and MSF [8], LiSTM demonstrates clear advancements in vehicle and cyclist detection which is attributed to motion-based feature integration. More details and discussions can be found in the Appendix.

In Figure 5, we compare the baseline with our module. LiSTM demonstrates superior capability, particularly highlighted by the pink arrows, in detecting cases that CenterPoint fails to identify due to distance and occlusion challenges. Additionally, LiSTM offers an increased number of positive samples with no annotations, as indicated by the yellow arrows.

On the nuScenes dataset, LiSTM outperforms the benchmarks PointPillar [11] and VoxelNet [37], improving NDS and mAP by 2-3% compared to CenterPoint [34]. Meanwhile, LiSTM is a boost in ATE and ASE as noted in Table 2.

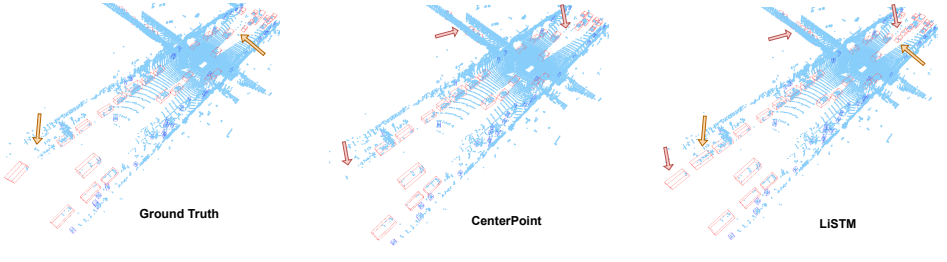

Figure 5: Qualitative visualization of our LiSTM on Waymo validation set. We show the 3D boxes predictions in the LiDAR bird's-eye-view

Ablation Study. As shown in Table 3, we compare the CenterPoint [34], Motion Transformer, Motion-Guided Feature Aggregation, and Dual Correlation Weighting Module sequentially for feature fusion structure, and we can see that CenterPoint is difficult to model multi-frame features. Meanwhile, modeling features solely through a Transformer can be

Experiment Number	Time	Veh. L2 APH	Ped. L2 APH	Cyl. L2 APH
1	t	62.13	60.91	61.16
2	$t-1,t$	63.41	58.17	62.37
3	$t-2,t-1,t$	63.46	58.62	63.89
4	$t-1,t,t+1$	**65.88**	61.10	**68.80**
5	$t-2,t-1,t,t+1,t+2$	65.73	**61.13**	67.56

Table 4: Ablation study of the frame fusion effects on Waymo validation set.

challenging. The proposed methods MGFA and DCWM offer a significant enhancement in APH by 2-3% through the incorporation of dynamic priors into the Transformer models.

Since our task is a multi-frame fusion strategy, we need to consider the number of frames to be used. In Table 4, we compare the effects of multi-frame fusion including single-frame, past-frame fusion, and past-future fusion. In summary, we can draw three key conclusions. Firstly, the fusion of cross-frame, as seen (EXP. 1,2, and 4), significantly contributes to detection results. Secondly, using too many frames (EXP. 5) not only increases memory requirements but also hampers model convergence. The main reason this conclusion differs from MSF [8] is that we use feature-level temporal fusion, whereas excessive attention stacking can hinder target characterization. Lastly, relying solely on past frames limits the model's understanding of the scene's geometry (EXP. 3 and 4). Incorporating both past and future frames provides a more comprehensive context for improved performance.

Motion Feature	Cyl. L2 APH	Veh. L2 APH
pre2cur	66.51	65.72
fut2cur	66.47	65.78
cur2pre	66.13	65.7
cur2fut	66.21	65.72
cur2pre + cur2fut	66.57	65.83
pre2cur + fut2cur	**68.80**	**65.88**

Table 5: Ablation study of motion-based heatmap feature selections on Waymo validation set.

Fusion Method	NDS	mAP
Concatenate	66.79	58.13
Attention [1]	65.37	58.99
Spatial Fusion [33]	67.13	60.31
DCWM	**68.32**	**63.77**

Table 6: Ablation study of different feature fusion strategies on Waymo validation set.

We select the motion feature as shown in Table 5, it fuses the information of object motion and encodes its features according to trajectory predictions. However, we find the feature observed at different times does not have much effect on the metrics. It can be concluded that the trajectory feature predicted by the future and the past for the present works best and is the most logical. For multiple frames feature map fusion, we sequentially compare the following schemes, concatenate, attention, and spatial-aware attention which are mentioned in CenterFormer [33] and our proposed DCWM in Table 6. We can discern that directly employing attention could hinder model learning, potentially yielding inferior results compared to concatenation and spatial fusion. However, our proposed Dual Correlation Weighting Module effectively fuses multiple frames and brings more pronounced enhancements.

5 Counclusion

Addressing the challenge of detecting sparse and occluded long-range LiDAR point clouds, we introduce LiSTM, a motion-based spatial-temporal fusion 3D point cloud detector. It leverages well-designed motion features and motion-guided feature fusion to enhance detection performance on Waymo and nuScenes datasets. In future work, we will focus on developing an end-to-end motion generator and exploring sparse feature representations.

Appendix

Computational Efficiency. We acknowledge that some reviewers have raised concerns regarding the computational resources. To address this, we compare CenterPoint, PVRCNN, MSF, and LiSTM. Despite LiSTM having significantly larger model parameters, its actual FPS is comparable to that of CenterPoint [34]. Moreover, LiSTM demonstrates a nearly 50% speed improvement over PV-RCNN++ [23] while consuming less memory and operating more efficiently than MSF [8]. This performance advantage primarily stems from our use of sparse feature operations and shared networks, which eliminate the need for computationally intensive processes such as multi-frame splicing and resampling.

Model	Model Parameter	Memory cost	FPS
CenterPoint [34]	7758811	2464 MiB	5.68 it/s
PV-RCNN++ [23]	13073505	3918 MiB	3.75 it/s
MSF [8]	15661651	6684 MiB	4.58 it/s
LiSTM	17592422	4400 MiB	5.26 it/s

Table 7: Computational efficiency

Point-Trajectory Model Analysis and Performance Comparison. Taking MSF [8] as an example, it enhances temporal features at the input level in two stages. In contrast, our approach targets implicit features, allowing for more efficient parallel computation and improved resource utilization. Unlike MSF's ROI sampling on point clouds, our method constructs a BEV heatmap, significantly boosting performance for larger targets like Vel (6m) and Cly (2m). However, for smaller targets like Ped (0.5m), even minor deviations can reduce performance, leading to lower results compared to MSF.

Lack Related Work on The Motion Estimation Model. Works [5, 29, 30] have proposed learnable SOT models and we will try to complete the end-to-end model in this direction in the future. However, this class of methods requires significant computational resources and is not well-suited for multiple target detectors. Therefore, our proposed strategy is to use a simple linear Kalman model for target trajectory prediction, which characterizes target motion a priori without the need for learnable parameters or GPU resources.

Total Waymo Evaluation. The model validation results for Waymo's full training dataset are shown below, focusing on a comparison between CenterPoint [34] and PVRCNN++ [23].

Model	Vehicle (AP/APH)↑		Pedestrian (AP/APH)↑		Cyclist (AP/APH)↑	
	L1	L2	L1	L2	L1	L2
CenterPoint [34]	72.64 / 72.10	64.57 / 64.07	74.53 / 68.36	66.50 / 60.84	71.14 / 69.91	68.56 / 67.37
PV-RCNN++ [23]	77.80 / 77.34	69.43 / 69.01	80.00 / 73.94	71.62 / 65.97	72.43/ 71.35	69.79 / 68.74
LiSTM	**78.91 / 78.31**	**70.64 / 70.10**	**80.79 / 75.01**	**72.16 / 66.87**	**74.42 / 73.33**	**71.84 / 70.79**

Table 8: Quantative comparison on Waymo validation set.

Long Distance Perception. The LiSTM architecture leverages continuous frames and motion priors to enhance performance, particularly for long-range detection. In our evaluation with the Waymo dataset, which covers a 75m radius horizontally and vertically, we use three distance thresholds to metric. Results show that LiSTM outperforms the baseline by an average of 5 points in the 25m to 75m range. Even beyond this range, where the point cloud is mostly filtered out, LiSTM metrics remain somewhat elevated compared to the baseline.

Model	25m away mAP↑			50m-75m mAP↑			75m away mAP↑		
	Vehicle	Pedestrians	Cyclist	Vehicle	Pedestrians	Cyclist	Vehicle	Pedestrians	Cyclist
CenterPoint [34]	58.80	63.12	61.37	41.82	54.00	50.50	11.46	16.30	14.82
LiSTM	**64.14**	**68.05**	**65.25**	**46.31**	**57.51**	**53.87**	**12.89**	**17.25**	**15.16**

Table 9: Long distance perception metric on the Waymo validation set.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[2] Ernesto Lozano Calvo, Bernardo Taveira, Fredrik Kahl, Niklas Gustafsson, Jonathan Larsson, and Adam Tonderski. Timepillars: Temporally-recurrent 3d lidar object detection. *arXiv preprint arXiv:2312.17260*, 2023.

[3] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *European Conference on Computer Vision*, pages 680–697. Springer, 2022.

[4] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023.

[5] Yubo Cui, Zhiheng Li, and Zheng Fang. Sttracker: Spatio-temporal tracker for 3d single object tracking. *IEEE Robotics and Automation Letters*, 2023.

[6] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Chenhang He, Ruihuang Li, Yabin Zhang, Shuai Li, and Lei Zhang. Msf: Motion-guided sequential fusion for efficient 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5196–5205, 2023.

[9] Kuan-Chih Huang, Weijie Lyu, Ming-Hsuan Yang, and Yi-Hsuan Tsai. Ptt: Point-trajectory transformer for efficient temporal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14938–14947, 2024.

[10] Aleksandr Kim, Aljoša Ošep, and Laura Leal-Taixé. Eagermot: 3d multi-object tracking via sensor fusion. In *2021 IEEE International Conference on Robotics and Automation*, pages 11315–11321. IEEE, 2021.

[11] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[12] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision*, pages 734–750, 2018.

[13] Yingwei Li, Charles R Qi, Yin Zhou, Chenxi Liu, and Dragomir Anguelov. Modar: Using motion forecasting for 3d object detection in point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9329–9339, 2023.

[14] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1477–1485, 2023.

[15] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[17] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[19] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2022.

[20] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[21] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, and Hongsheng Li. Part-a2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670*, 2(3), 2019.

[22] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[23] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023.

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset, 2020.

[25] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. GitHub-open-mmlab/OpenPCDet: OpenPCDetToolboxforLiDAR-based3DObjectDetection., 2020.

[26] Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li, and Zize Liang. Apple detection during different growth stages in orchards using the improved yolo-v3 model. *Computers and Electronics in Agriculture*, 157:417–426, 2019.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3631, 2023.

[29] Yan Xia, Qiangqiang Wu, Wei Li, Antoni B Chan, and Uwe Stilla. A lightweight and detector-free 3d single object tracker on point clouds. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5543–5554, 2023.

[30] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10448–10457, 2021.

[31] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[32] Zetong Yang, Yin Zhou, Zhifeng Chen, and Jiquan Ngiam. 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1863–1872, 2021.

[33] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11495–11504, 2020.

[34] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021.

[35] Runkai Zhao, Yuwen Heng, Heng Wang, Yuanda Gao, Shilei Liu, Changhao Yao, Jiawen Chen, and Weidong Cai. Advancements in 3d lane detection using lidar point clouds: From data collection to model development. In *2024 IEEE International Conference on Robotics and Automation*, pages 5382–5388. IEEE, 2024.

[36] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[37] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[38] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *European Conference on Computer Vision*, pages 496–513. Springer, 2022.