MSF: Motion-guided Sequential Fusion for Efficient 3D Object Detection from Point Cloud Sequences

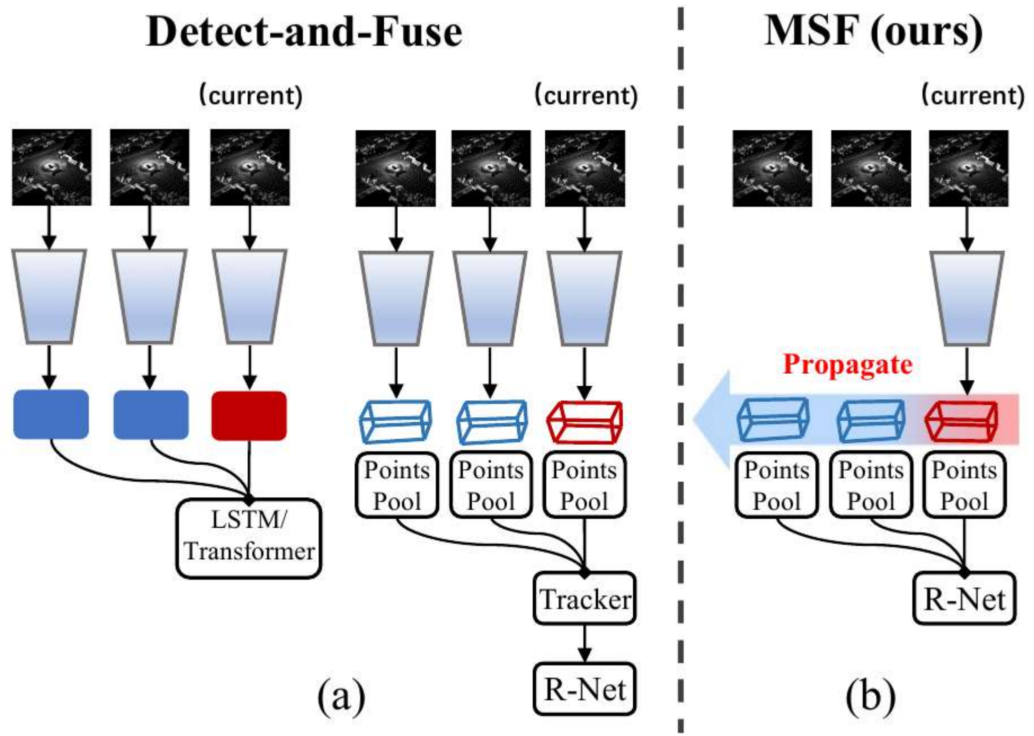
Paper Tag: TUE-PM-101

Chenhang He, Ruihuang Li, Yabin Zhang, Shuai Li, Lei Zhang

The Hong Kong Polytechnic University

Code: https://github.com/skyhehe123/VoxSeT

Motivation


Detect-and-Fuse
MSF (ours)
(current)
(current)
(current)
LSTM/
Transformer
Points
Pool
Tracker
R-Net
Propagate
(a)
(b)

Motion-guided Sequential Fusion

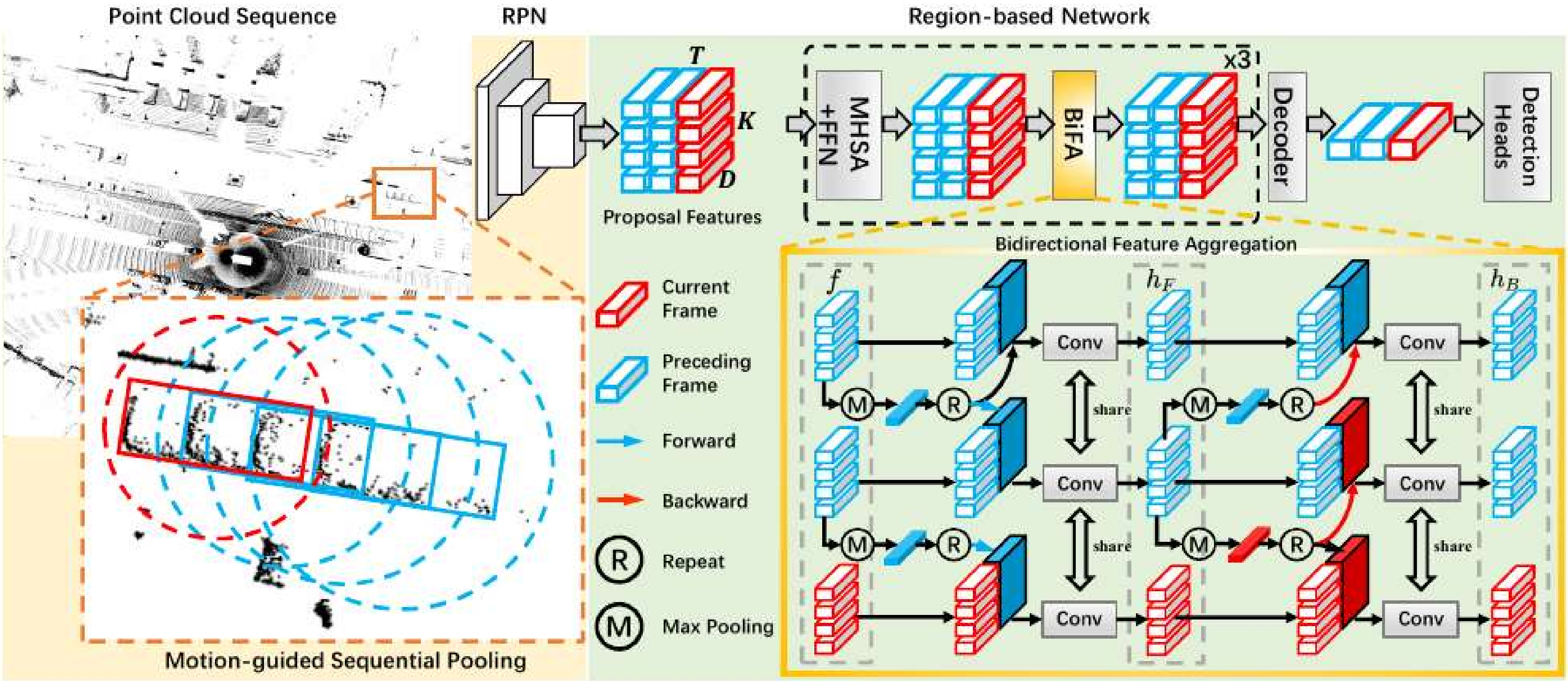
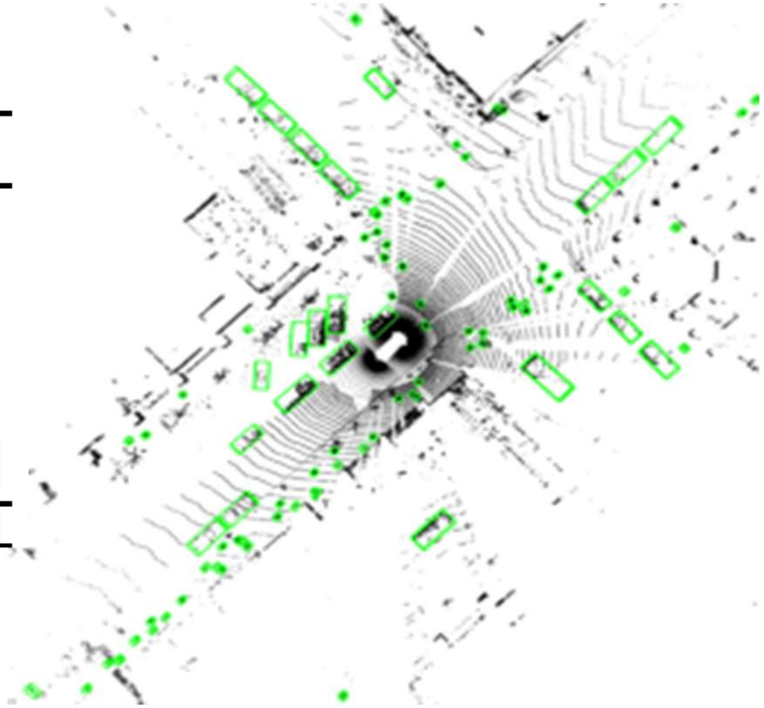


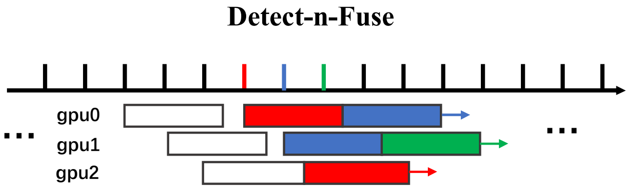
Table 4. Performance comparison on the test set of Waymo Open Dataset.

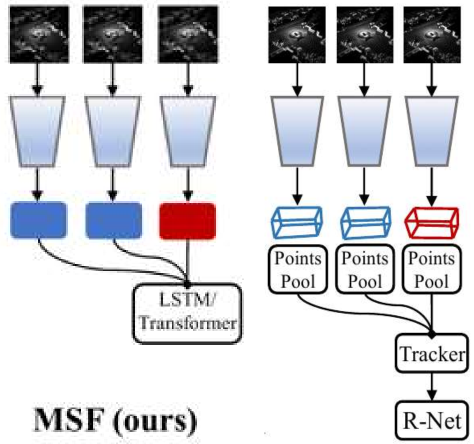
Method	ALL (3D mAPH)		Vehicle (AP/APH)		Pedestrian (AP/APH)		Cyclist (AP/APH)	
	L1	L2	L1	L2	L1	L2	L1	L2
PointPillar [9]	-	-	68.10	60.10	68.00/55.50	61.40/50.10	-	-
StarNet [15]	-	-	61.00	54.50	67.80/59.90	61.10/54.00	-	-
M3DETR [5]	67.1	61.9	77.7/77.1	70.5/70.0	68.2/58.5	60.6/52.0	67.3/65.7	65.3/63.8
3D-MAN [31]	-	-	78.28	69.98	69.97/65.98	63.98/60.26	-	-
PV-RCNN++ [22]	75.7	70.2	81.6/81.2	73.9/73.5	80.4/75.0	74.1/69.0	71.9/70.8	69.3/68.2
CenterPoint [33]	77.2	71.9	81.1/80.6	73.4/73.0	80.5/77.3	74.6/71.5	74.6/73.7	72.2/71.3
RSN [26]	-	-	80.30	71.60	78.90/75.60	70.70/67.80	-	-
SST-3f [3]	78.3	72.8	81.0/80.6	73.1/72.7	83.3/79.7	76.9/73.5	75.7/74.6	73.2/72.2
MPPNet [2]	80.59	75.67	84.27/83.88	77.29/76.91	84.12/81.52	78.44/75.93	77.11/76.36	74.91/74.18
CenterFormer [37]	80.91	76.29	85.36/84.94	78.68/78.28	85.22/ 82.48	80.09/77.42	76.21/75.32	74.04/73.17
MSF (ours)	**81.74**	**76.96**	**86.07/85.67**	**79.20/78.82**	**85.99/83.10**	**80.61/77.82**	**77.29/76.44**	**75.09/74.25**



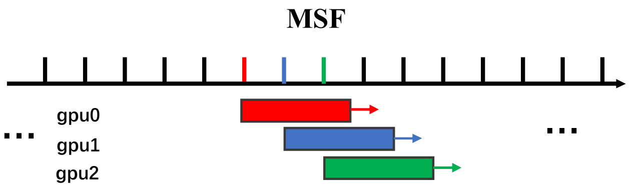
Motivation

- The ``Detect-and-Fuse‘’ framework
  - Redundant computation on background
  - Introduce congestion and latency if $T_{net} > T_{data}$

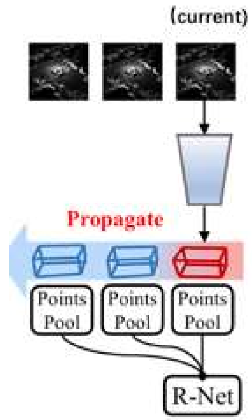




- MSF (ours)
  - Reuse the region-of-interest in preceding frames
  - Be efficient as a single-frame detector



MSF (ours)



Motion-guided Sequential Pooling

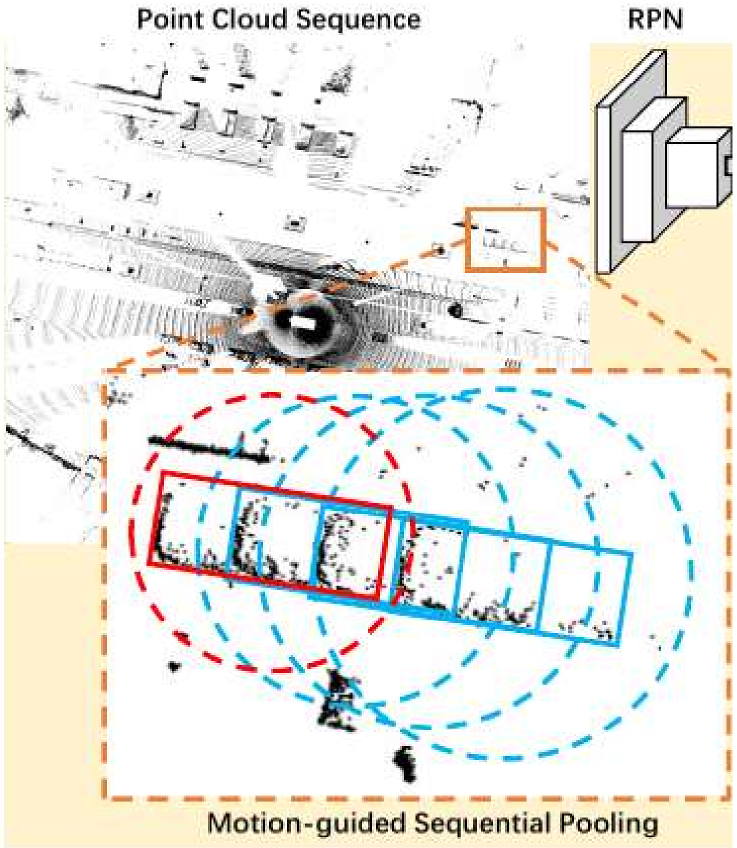
- Pooling by propagating the proposals generated on the current frame to preceding frames based on their estimated velocities (v_x, v_y)

$$(x^t - p_x + v_x \cdot \Delta t)^2 + (y^t - p_y + v_y \cdot \Delta t)^2 < (\frac{d^t}{2})^2,$$

- Geometric & Motion Encoding

$$g_i^t = \mathrm{MLP}(\mathcal{S}(\{l_i^t - b_j^t\}_{j=0}^{8})), \text{for } i = 1, ..., K,$$

$$m_i^t = \mathrm{MLP}(\mathrm{Concat}(\{l_i^t - b_j^0\}_{j=0}^{8}, \Delta t)), \text{for } i = 1, ..., K.$$

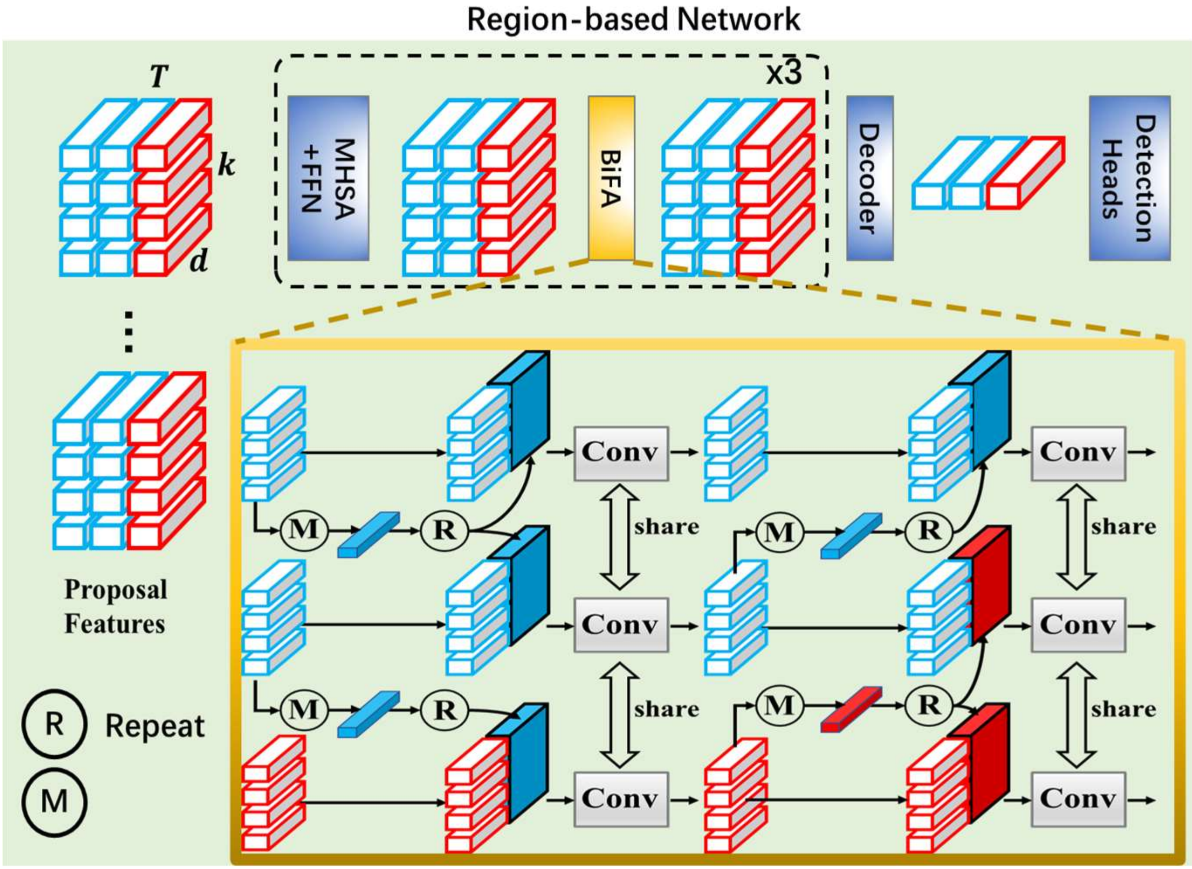


Region-based Network

- Intra-frame Fusion
  - Self-Attention
  - FFN
- Cross-frame Fusion
  - Bidirectional Feature Aggregation

$$h_F^t = \mathrm{Conv}(\mathrm{Concat}(f^t, \mathrm{Repeat} \circ \mathrm{Max\text{-}pool}(f^{t-1})))$$

$$h_B^t = \mathrm{Conv}(\mathrm{Concat}(h_F^t, \mathrm{Repeat} \circ \mathrm{Max\text{-}pool}(h_F^{t+1})))$$



Efficient Pooling with Voxel-Sampling

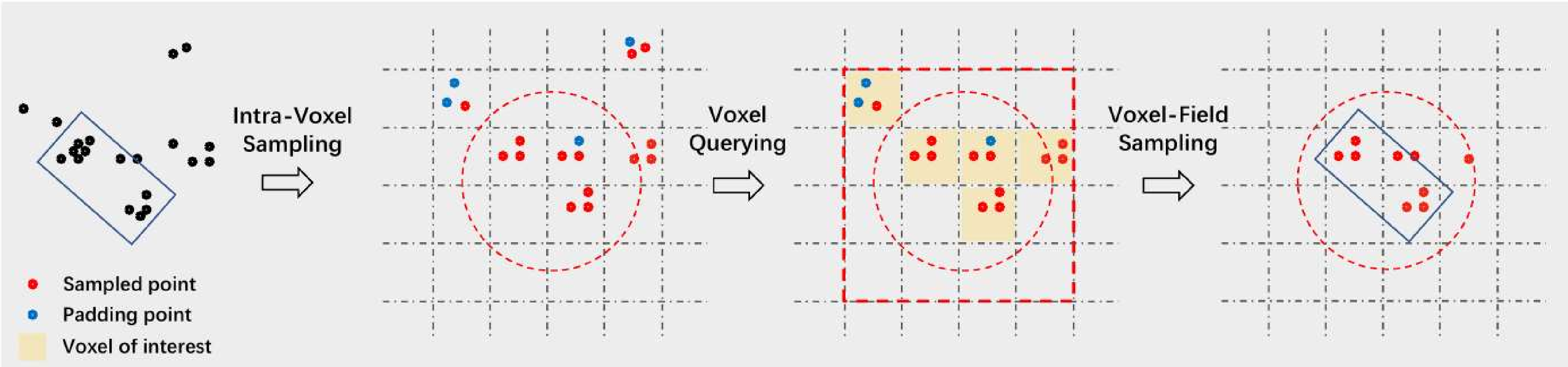


Figure 3. Illustration of our optimized point cloud pooling method. We first perform intra-voxel sampling to keep a fixed number of points in each voxel. Then we query $n \times n$ voxels fields for each proposal and uniformly draw points from the non-empty voxels within.

Table 2. The latency of point cloud pooling on 1-frame, 4-frames and 8-frames sequences.

	N=168k	N=674k	N=1382k
Cylindrical Pooling	8.2ms	25.2ms	40.1ms
Our Optimized	2.3ms	3.4ms	5.0ms

Experiments

Table 3. Performance comparison on the validation set of Waymo Open Dataset.

Method	Frames	ALL (3D mAPH)		Vehicle (AP/APH)		Pedestrian (AP/APH)		Cyclist (AP/APH)	
		L1	L2	L1	L2	L1	L2	L1	L2
SECOND [28]	1	63.05	57.23	72.27/71.69	63.85/63.33	68.70/58.18	60.72/51.31	60.62/59.28	58.34/57.05
PointPillar [9]	1	63.33	57.53	71.60/71.00	63.10/62.50	70.60/56.70	62.90/50.20	64.40/62.30	61.90/59.90
IA-SSD [35]	1	64.48	58.08	70.53/69.67	61.55/60.80	69.38/58.47	60.30/50.73	67.67/65.30	64.98/62.71
LiDAR R-CNN [10]	1	66.20	60.10	73.50/73.00	64.70/64.20	71.20/58.70	63.10/51.70	68.60/66.90	66.10/64.40
RSN [26]	1	-	-	75.10/74.60	66.00/65.50	77.80/72.70	68.30/63.70	-	-
PV-RCNN [21]	1	69.63	63.33	77.51/76.89	68.98/68.41	75.01/65.65	66.04/57.61	67.81/66.35	65.39/63.98
Part-A2 [24]	1	70.25	63.84	77.05/76.51	68.47/67.97	75.24/66.87	66.18/58.62	68.60/67.36	66.13/64.93
Centerpoint [33]	1	-	65.50	-	-/66.20	-	-/62.60	-	-/67.60
VoTR [14]	1	-	-	74.95/74.25	65.91/65.29	-	-	-	-
VoxSeT [6]	1	72.24	66.22	74.50/74.03	65.99/65.56	80.03/72.42	72.45/65.39	71.56/70.29	68.95/67.73
SST-1f [3]	1	-	-	76.22/75.79	68.04/67.64	81.39/74.05	72.82/65.93	-	-
SWFormer-1f [25]	1	-	-	77.8/77.3	69.2/68.8	80.9/72.7	72.5/64.9	-	-
PillarNet [20]	1	74.60	68.43	79.09/78.59	70.92/70.46	80.59/74.01	72.28/66.17	72.29/71.21	69.72/68.67
PV-RCNN++ [22]	1	75.21	68.61	79.10/78.63	70.34/69.91	80.62/74.62	71.86/66.30	73.49/72.38	70.70/69.62
3D-MAN [31]	16	-	-	74.53/74.03	67.61/67.14	71.7/67.7	62.6/59.0	-	-
SST-3f [3]	3	-	-	78.66/78.21	69.98/69.57	83.81/80.14	75.94/72.37	-	-
SWFormer-3f [25]	3	-	-	79.4/78.9	71.1/70.6	82.9/79.0	74.8/71.1	-	-
CenterFormer [37]	4	77.0	73.2	78.1/77.6	73.4/72.9	81.7/78.6	77.2/74.2	75.6/74.8	73.4/72.6
CenterFormer [37]	8	77.3	73.7	78.8/78.3	74.3/73.8	82.1/79.3	77.8/75.0	75.2/74.4	73.2/72.3
MPPNet [2]	4	79.83	74.22	81.54/81.06	74.07/73.61	84.56/81.94	77.20/74.67	77.15/76.50	75.01/74.38
MPPNet [2]	16	80.40	74.85	82.74/**82.28**	75.41/74.96	84.69/82.25	77.43/75.06	77.28/76.66	75.13/74.52
MSF (ours)	4	80.20	74.62	81.36/80.87	73.81/73.35	85.05/82.10	77.92/75.11	78.40/77.61	76.17/75.40
MSF (ours)	8	**80.65**	**75.46**	**82.83**/82.01	**75.76/75.31**	**85.24/82.21**	**78.32/75.61**	**78.52/77.74**	**76.32/75.47**

Experiments

Table 4. Performance comparison on the test set of Waymo Open Dataset.

Method	ALL (3D mAPH)		Vehicle (AP/APH)		Pedestrian (AP/APH)		Cyclist (AP/APH)	
	L1	L2	L1	L2	L1	L2	L1	L2
PointPillar [9]	-	-	68.10	60.10	68.00/55.50	61.40/50.10	-	-
StarNet [15]	-	-	61.00	54.50	67.80/59.90	61.10/54.00	-	-
M3DETR [5]	67.1	61.9	77.7/77.1	70.5/70.0	68.2/58.5	60.6/52.0	67.3/65.7	65.3/63.8
3D-MAN [31]	-	-	78.28	69.98	69.97/65.98	63.98/60.26	-	-
PV-RCNN++ [22]	75.7	70.2	81.6/81.2	73.9/73.5	80.4/75.0	74.1/69.0	71.9/70.8	69.3/68.2
CenterPoint [33]	77.2	71.9	81.1/80.6	73.4/73.0	80.5/77.3	74.6/71.5	74.6/73.7	72.2/71.3
RSN [26]	-	-	80.30	71.60	78.90/75.60	70.70/67.80	-	-
SST-3f [3]	78.3	72.8	81.0/80.6	73.1/72.7	83.3/79.7	76.9/73.5	75.7/74.6	73.2/72.2
MPPNet [2]	80.59	75.67	84.27/83.88	77.29/76.91	84.12/81.52	78.44/75.93	77.11/76.36	74.91/74.18
CenterFormer [37]	80.91	76.29	85.36/84.94	78.68/78.28	85.22/ 82.48	80.09/77.42	76.21/75.32	74.04/73.17
MSF (ours)	**81.74**	**76.96**	**86.07/85.67**	**79.20/78.82**	**85.99/83.10**	**80.61/77.82**	**77.29/76.44**	**75.09/74.25**

Discussion

- Propagated proposals have the same size over the sequence, thus avoiding the use of proxy points to maintain a consistent representation over the sequence.

- Raw point-based features can achieve higher accuracy with self-attention layers.

Config	Vehicle.	Pedestrian.	Cyclist
Raw + SA	73.35	75.11	75.40
Proxy + SA	73.12 (-0.23)	74.20 (-0.91)	74.32 (-1.08)
Proxy + Mixer	73.45(+0.10)	74.13 (-0.98)	74.39 (-1.01)

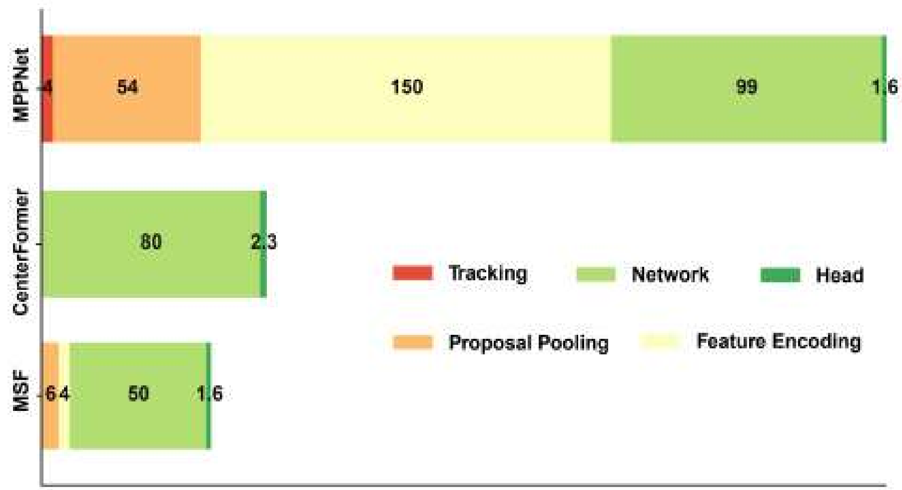


Figure 4. Comparison of the runtime of different methods.

Table 9. The runtime decomposition of MPPNet and MSF.

MPPNet		MSF	
MLP-Mixer	Cross-Attention	Self-Attention	BiFA
75 ms	24 ms	36 ms	12 ms

Conclusion

- An efficient Motion-guided Sequential Fusion (MSF) method is proposed to fuse multi-frame point clouds at region level by propagating the proposals of current frame to preceding frames based on the object motions.
- A novel Bidirectional Feature Aggregation (BiFA) module is introduced to facilitate the interactions of proposal features across frames.
- The point cloud pooling method is optimized with a voxel-based sampling technique, which significantly reduces the runtime on large-scale point cloud sequence.

The code is available at **https://github.com/skyhehe123/MSF**