Sequential Point Clouds: A Survey

Haiyan Wang and Yingli Tian, *Fellow, IEEE,*

Abstract—Point clouds have garnered increasing research attention and found numerous practical applications. However, many of these applications, such as autonomous driving and robotic manipulation, rely on sequential point clouds, essentially adding a temporal dimension to the data (i.e., four dimensions) because the information of the static point cloud data could provide is still limited. Recent research efforts have been directed towards enhancing the understanding and utilization of sequential point clouds. This paper offers a comprehensive review of deep learning methods applied to sequential point cloud research, encompassing dynamic flow estimation, object detection & tracking, point cloud segmentation, and point cloud forecasting. This paper further summarizes and compares the quantitative results of the reviewed methods over the public benchmark datasets. Ultimately, the paper concludes by addressing the challenges in current sequential point cloud research and pointing towards promising avenues for future research.

Index Terms—4D sequential point cloud; Deep learning; Flow estimation; Object detection & tracking; Point cloud segmentation; Point cloud forecasting.

1 Introduction

With the development of recent deep learning and sensor technologies, the expense of 3D point cloud acquisition has significantly dropped. Point cloud data can be easily captured through 3D scanners, Lidars, or RGBD cameras, which comprise a set of unordered points represented by *XYZ* in world coordinates with permutation invariant properties. Compared to other data formats, like 2D image, even 3D voxel or mesh, the point cloud is a more practical data representation for our real world. 2D images lose the spatial geometric information of 3D space, while other grid-based representations (e.g. voxel and mesh) suffer from the redundancy of the inner space representation and massive computation.

Recent research efforts have made great contributions to the static point cloud learning process. The survey paper [32] provided an elaborate summary of 3D point cloud learning methods including various downstream tasks and applications. Basically, some methods just pursued an easier deep learning way which employs the convolution operation on the high dimension 3D data [59], [113], [145]. These methods usually require transferring point cloud to other regular data formats such as voxel or mesh representations. The input of grid data representation makes it possible to extend the idea of advanced 2D convolution network design to the 3D domain for high-level feature extraction. Although the convolution is attractive, these methods suffer a lot from heavy computation costs and quantization errors due to the grid representation. The seminal work PointNet [80] and PointNet++ [81] introduced a straightforward solution based on raw point cloud input and extracted high-level feature representations through novel sampling and grouping strategies. Inspired by these two pioneer methods, tremendous studies developed more and more advanced structures and achieved impressive performance on different computer vision applications. These direct point-based methods usually maximally preserved the 3D geometry information of the input data and well balanced the efficiency and efficacy.

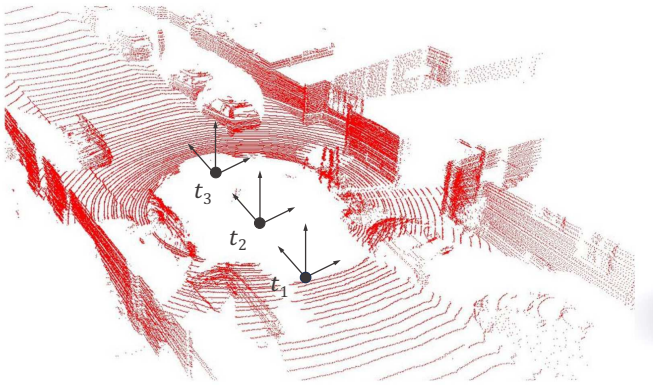


Fig. 1: Demonstration of sequential point cloud stacked by a sequence of point cloud frames. The figure is from [119] with author's permission.

However, static point cloud is limited to fully represent our real world especially when there are motions. The dynamic real world is actually with three spatial dimensions plus one time dimension (i.e. 4D), which leads to a huge uncertainty compared to the single static point cloud. The features of the scene or objects may change along the time sequence causing the potential missing, occlusion, or unseen information. Even these uncertainties are inevitable in our dynamic world, it is critical to be aware of them and estimated especially in real-world applications such as self-driving or AR/VR techniques. Thus, many deep learning tasks (e.g. dynamic flow estimation, object detection & tracking, point cloud segmentation, and point cloud forecasting, etc.) are worth to explore for learning the spatio-temporal information from 4D sequential point cloud (SPL) data. In a short period, the motion information such as point flow which is similar to 2D optical flow can be estimated based on consecutive point cloud frames. Also, based on the previous several frames, the point cloud of the future moment can be predicted which is applied by vast kinds of forecasting tasks such motion

- *Haiyan Wang is with the Department of Electrical Engineering, The City College of New York, New York, NY, 10031. E-mail: hwang3@ccny.cuny.edu*
- *Yingli Tian (Corresponding author) is with the Department of Electrical Engineering, The City College, and the Department of Computer Science, the Graduate Center, the City University of New York, New York, NY, 10031. E-mail: ytian@ccny.cuny.edu*

This material is based upon work supported by the National Science Foundation under award number IIS-2041307.

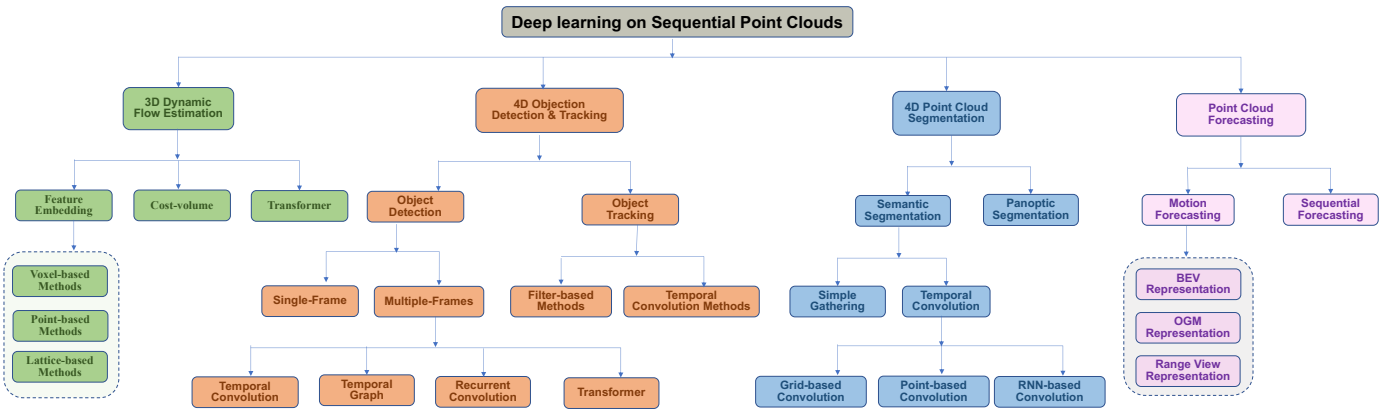


Fig. 2: A taxonomy of deep learning methods for sequential point cloud.

and sequential forecasting. The point cloud generation falls in this application as well. The recent thriving tasks such as object tracking, action recognition, 4D point cloud reconstruction, and even the 4D segmentation can also benefit from the long-time temporal information embedded in the point cloud sequence. Motivated by the distinguished property of the sequential point cloud and these popular applications, the research focuses are diverting from the static point cloud to the dynamic sequential point cloud.

Sequential point cloud (SPL), as shown in Figure 1, is defined as a sequence of static point cloud frames $S = S_1, S_2, ...S_t, ..., S_T, (t = 1, 2, ..., T)$ where T is the time length. Each point cloud frame S_t consists of a set of unordered points which are permutation invariant $S_t = p_1, p_2, ..., p_n, ..., p_N$, N is the number of points for the point cloud frame S_t. The point p_n inside S_t is represented with both 3D location $X_n \in \mathbb{R}^3$ and feature vector $F_n \in \mathbb{R}^c$. Compared to static point cloud, SPL is unique with the following properties:

- **Large scale**. A static scene point cloud normally contains plenty of points and can easily reach a scale of millions. SPL unites a sequence of static point clouds, the number of points are extremely immense.
- **Permutation invariant of single frame**. Every single scan in SPL is a set of unordered points which is invariant to any permutation and geometric transformation such as translation or rotation. These operations will not alter the point cloud properties or classification results.
- **Permutation variant for multiple frames**. Among multiple frames of point clouds, the order of these frames is the most critical characteristic which makes it distinctive. It reflects the temporal information along with the time series including the dynamic motion and deformation of the object in the point clouds.
- **4D Contextual Correlation / Continuum**. The learning of SPL ought not to separate the spatial and the temporal. Instead, for the 4D continuum, a spatio-temporal correlation structure contains extremely rich contextual information availing a better scene understanding compared to the single static point cloud.

Despite the superior properties and importance of SPL, it is especially challenging to process 4D data in an effective and efficient manner due to the large scale and sophistication of the spatio-temporal relations between multiple frames. To optimally represent 4D data, numerous embedding techniques are developed for processing point cloud inputs. These methods can be integrated with diverse network architectures or tailored to specific computer vision tasks, ensuring a comprehensive and effective data representation. The core idea of processing 4D sequential point cloud data is to take benefit of both spatial and temporal dimensions. Meanwhile, the way of extracting and merging temporal information is essential during this process. Many methods have been developed, showcasing remarkable performance when applied to static 3D point clouds.

Some previous reviews have provided summaries of deep learning methods for general 3D data [3], [40], [83], [128] or especially to the static point cloud [32], [55]. However, none of them focus on modeling SPL. This paper presents an extensive review of the deep learning-based methods for 4D SPL research and emphasizes the temporal encoding and modeling of the spatio-temporal correlation structure. As shown in Figure 2, we provide a thorough comparison of existing methods on public benchmark datasets, covering a wide range of tasks and applications including dynamic flow estimation, object detection & tracking, point cloud segmentation, and point cloud forecasting. Additionally, we offer a concise summary of the research challenges of SPL and highlight several emerging trends that warrant attention in future research.

The rest of the paper is organized as follows. Sec. 2 introduces the common point cloud embedding to represent SPL data. The downstream tasks of different applications of SPL are summarized in Sec. 3 for scene flow estimation, Sec. 4 for objection detection, Sec. 5 for object tracking, Sec. 6 for object segmentation, and Sec. 7 for point cloud forecasting. Sec. 8 provides a few potential future research directions on SPL and Sec. 9 concludes the whole survey. The descriptions of the commonly used deep network architectures and datasets for SPL can be found in the attached Supplementary.

The primary objective of this survey is to offer a comprehensive overview of the predominant techniques employed in processing sequential point clouds. Given the vast array of existing methods in this field, it is impractical to cover each one exhaustively. Therefore, a deliberate selection was made to identify and focus on a representative subset of methods. This subset is chosen to encompass a diverse range of approaches, ensuring that the survey provides a broad perspective on the various types of methodologies used in this area of study.

2 Common Point Cloud Embedding

Various network architectures are intimately bound up with distinct embeddings for point clouds. While these networks share the

common objective of extracting meaningful information from 3D point cloud data, their designs can vary significantly based on the chosen data embedding. Current approaches can be categorized into three classes: point-based, grid-based, and implicit neural embeddings.

2.1 Point-based Embedding

In point-based representations, each point in the point cloud is treated as an independent entity, and features are computed directly from the coordinates and attributes of these points. The pioneering work PointNet [80] proposed a general architecture directly taking point cloud data as input and extracting global 3D features. Thus the network is capable of digesting the unordered point sets while being invariant to permutations of point order. The following work PointNet++ [81] extended PointNet's architecture to capture hierarchical and multi-scale features, further advanced the concept of point-based embeddings. Their architectures and concepts have influenced subsequent research [13], [46], [51], [53], [58], [103], [127], shaping the development of more advanced point-based embedding methods and enabling a wide range of applications in 3D data processing.

Essentially, point-based embedding offers several distinct advantages when processing 3D data. First and foremost, they preserve the fine-grained details of the 3D data due to the point-wise dense representation. Additionally, they are invariant to the order of points, which makes them especially well-suited for handling unstructured and irregularly sampled point clouds without the need for any pre-processing. Notably, these methods can be computationally efficient with sparse point clouds since they focus solely on processing relevant points rather than entire volumetric grids.

On the flip side, there are inherent challenges with point-based embedding. Large point clouds or intricate architectures can exert substantial computational demands on such methods, requiring heavy computational resources for both the training and inference phases. Another challenge is that variations in point density can affect the performance of these embeddings, particularly when handling irregularly sampled data. Unlike their grid-based counterparts, point-based embedding lacks a natural structured grid, which may impact certain tasks like convolutions.

2.2 Grid-based Embedding

Grid-based embedding is a powerful tool for 3D data analysis, making it easier to handle and compute point cloud data. The method breaks down the 3D space into regular grid cells or voxels, treating each cell as a mini-region within the larger 3D space [23], [45], [62], [106], [127]. Features from the points in each cell are extracted using a mix of techniques like convolutions, pooling, and other aggregation methods. These techniques help capture fine-grained details about each point, making grid-based methods effective for both local and global spatial analysis. Because of these strengths, grid-based embedding is especially useful for tasks that rely on understanding spatial relationships, such as 3D object detection, segmentation, and occupancy mapping. These approaches also bridge traditional image-based Convolutional Neural Networks (CNNs) and point-based methods, widening the toolkit for 3D data processing.

However, grid-based embedding is not a one-size-fits-all solution. On the upside, it is computationally efficient, making it suitable for real-time applications. It also meshes well with CNNs for feature extraction in machine learning. Transforming point clouds into a grid also compresses the data, reducing both storage and computational costs. Plus, the grid's structure helps average out any noise, making the data more reliable. But there are trade-offs. The choice of grid resolution—fine or coarse—affects both computational performance and detail capture. High-res grids offer more detail but can be a drain on resources, while low-res grids are faster but might miss important features. Also, converting points to a grid format could mean losing some original point-based details, which could be a problem for some applications such as point cloud compression or fine-grained 3D reconstruction.

2.3 Implicit Neural Embedding

Implicit neural embedding [26], [42], [54], [68], [76], [89], [133] represents an alternative category of machine learning techniques designed to encode and manipulate 3D geometric structures within 3D point clouds. Instead of explicitly storing the coordinates of each point, face, or voxel, it typically uses neural networks to implicitly define the surface or volume of an object in 3D space. Specifically, a common approach for implicit neural embedding uses a conditional neural network that takes 3D coordinates (x, y, z) as input and outputs a scalar value, usually interpreted as the "occupancy probability" or "signed distance function" at that coordinate. In this way, the entire 3D object can be implicitly represented through this neural network model, significantly reducing the costs of storage and computation. Most importantly, this representation is robust to the noise of point cloud origin location, orientation and the 3D coordinate system. Potential application of sequential point cloud implicit embedding can be found in Sec. 8.

Nevertheless, while having numerous advantages, one notable drawback is the training complexity, which can be computationally extensive and time-consuming, particularly for high-resolution 3D models, potentially leading to scalability issues for larger or more detailed models. The accuracy of these models can sometimes be compromised, especially around the detailed regions, leading to a potential loss of detail or inaccuracies in the reconstruction. Additionally, there may be surface ambiguities that could cause difficulties in precisely determining object boundaries.

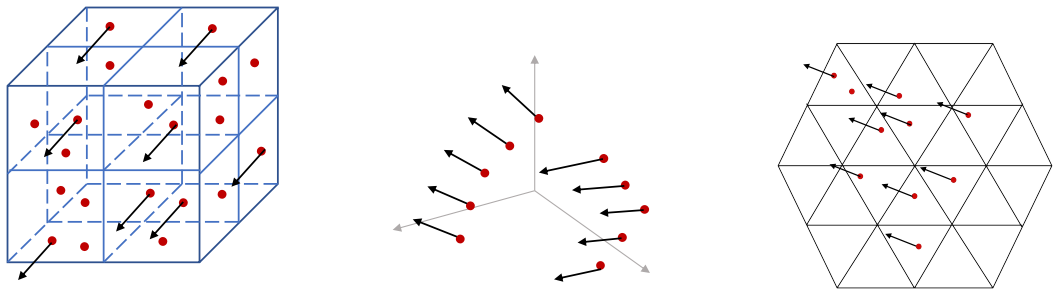
3 Scene Flow Estimation

In dynamic SPL, scene flow estimation is one of the most crucial and fundamental tasks. It is playing an important role in the applications of robotics manipulation, autonomous driving etc. Flow actually describes the motion status of objects. Specifically in 4D point cloud, scene flow demonstrates 3D velocity of each 3D point in a scene. Assuming there are two consecutive point clouds in a point cloud sequence $S_t = \{p_i^t, i = 1, 2, ..., N_t\}$, and $S_{t+1} = \{p_i^{t+1}, i = 1, 2, ..., N_{t+1}\}$, scene flow $D_t = \{D_i^t, i = 1, 2, ..., N_t\}$ is defined as the translation motion vector between S_t and S_{t+1}. For each point p_i^t in S_t, the translated point is defined as q_i^{t+1}. $D_i^t = q_i^t - p_i^t$. It worth to note that q_i^t and p_i^{t+1} are not necessary to be the same location.

Here we categorize the existing point cloud scene flow estimation methods into *feature embedding-based*, *cost-volume-based*, and *transformer-based* methods. A list of scene flow estimation methods can be found in Table 1.

3.1 Feature Embedding-based Methods

Feature embedding methods for scene flow estimation aim to derive compact representations from sequential point cloud data,



(a) Voxel-based representation. (b) Point-based representation. (c) Lattice-based representation.

Fig. 3: The illustration of different representations for scene flow estimation methods. Red points show the first point cloud frame while black arrows demonstrate related scene flow vectors.

capturing the essence of both spatial structures and temporal dynamics. By transforming raw data points into higher-level features, these methods can efficiently track and predict motion patterns across consecutive frames, ensuring more accurate scene flow predictions while minimizing computational overhead. Their strength lies in discerning subtle changes over time, enabling a deeper understanding of scene dynamics and motion trajectories in 3D environments.

3.1.1 Voxel-based Methods

These methods [5], [44], [74] convert SPL into a volumetric representation for motion feature extraction using 3D CNNs. Scene flow is calculated from voxel centroids, as depicted in Figure 3a. Initially, the input point cloud is segmented into voxels, processed through networks like VoxelNet [145]. PointFlowNet [5] predicts 3D object boundaries and their motion, using multiple decoder branches for scene flow, ego-motion, and object detection. It integrates scene flow with object detection for pixel and object level motion analysis, optimized through combined loss functions. VoxFlowNet [74], similar to PointFlowNet, utilizes voxel representation for scene flow estimation. It differs in its point selection, using the farthest point sampling strategy, and integrates PointNet++ [81] and FlowNet3D [56] concepts. VoxFlowNet aggregates local neighbor features in each voxel, employs Set Conv layers for feature extraction, and Set Upconv layers for upscaling voxels to original scale for scene flow estimation.

However, existing methods struggle with large-scale point clouds due to computational demands. Scalable [44] addresses this by supporting point clouds up to O(100K) in real-time. It leverages PointPillars [49] for feature extraction, dynamic voxelization, and a U-Net autoencoder for processing, with shared MLP layers for point-wise scene flow prediction. This approach reduces computation significantly compared to KNN-based neighbor searches. Additionally, Scalable introduces a new benchmark for scene flow estimation using the Waymo Open Dataset [99], addressing the scarcity of real, annotated scene flow datasets.

3.1.2 Direct Point-based Methods

While voxel-based methods can suffer from redundancy and incomplete information, point-based approaches [56], [108], [112] address these issues by directly using raw point clouds to estimate scene flow vectors. As illustrated in Figure 3b, these methods compute a scene flow vector for each point, extracting features spatially and temporally.

FlowNet3D [56], a notable example, utilizes PointNet++ [81] for feature extraction from consecutive point cloud frames. It employs farthest point sampling for neighbor points and hierarchically aggregates local features. The flow embedding layer concatenates features of two frames, and the flow is refined through set upconv layers, leading to impressive performance on datasets like Flythings3D and KITTI Scene Flow 2015. Shao et al. [92] proposed a concurrent method that estimates scene flow alongside segmentation and motion trajectories using RGBD images, differing from FlowNet3D's reliance solely on point clouds. However, FlowNet3D's primary limitation was its use of simple $l2$ loss for comparing predicted and actual scene flows. FlowNet3D++ [112] builds upon its predecessor with two innovative loss functions: the point-to-plane loss, enhancing performance in dynamic scenes, and the cosine distance loss, correcting direction discrepancies in flow vectors. Furthermore, it introduces a 3D dynamic reconstruction pipeline, significantly improving performance over the original FlowNet3D with this new evaluation metric.

Almost all of the previous paper adopted PointNet++ [81] as their feature extraction backbone. However, one major issue related to PointNet++ is the irregular sampling which leads to the randomness for feature extraction process. FESTA [108] used a spatial-temporal attention mechanism and achieved prominent benefits for scene flow estimation benchmarks. In the spatial domain, FESTA exploited a novel SA^2 layer to extract those points which were more stable and critical. The more representative points tended to help the network find better correspondence between the continuous frames. Likewise, in the temporal domain, FESTA introduced a TA^2 layer to tackle the various motion scale problem. A recurrent design was employed to first estimate an initial flow. Afterward, in the second iteration, FESTA shifted the attended region based on the initial flow which had more likelihood to find the good matches. The extensive experiments exhibited the significance of the proposed attention mechanism on scene flow estimation task.

Pure point-based solution still concentrates on local correlations. The absence of global information leads to the error accumulation during previous coarse-to-fine strategies. Thus, the authors of [114] proposed a method named PV-RAFT applying point and voxel representations together to capture all-pairs correspondence. The K-NN pairs were adopted to model the local correspondence while pairs between volumes were utilized to involve global correlations. This improved the scene flow estimation performance especially for fast moving objects.

Just Go with the Flow [71] was another recent work that solely focused on using a unsupervised method and solving the lack of ground truth annotations in the real-world point cloud scene flow datasets. The authors built the network upon the FlowNet3D [56]

TABLE 1: The summary of scene flow estimation methods

Methods		Code	Attribute
Feature Embedding	PointFlowNet [5]	×	Feature Embedding based methods usually learn a compact, discriminative representation of the raw input data (like voxels or point clouds) via MLP or convolutional layers, allowing for a better understanding and representation of the scene and its dynamics.
	VoxFlowNet [74]	✓	
	Scalable [44]	✓	
	PV-RAFT [114]	✓	
	FlowNet3D [56]	✓	
	FlowNet3D++ [112]	×	
	FESTA [108]	✓	
	HPLFlowNet [31]	✓	
	Just Go [71]	✓	
Cost-volume	PointPWCNet [126]	✓	Cost-volume based methods compute the similarity between corresponding voxels, or points from two different frames or views in a coarse-to-fine manner.
	Res3DSF [107]	✓	
Transformer	PT-FlowNet [25]	✓	Transformer based methods are built upon the transformer layers to capture precise correlation between frames through attention mechanism.
	SCTN [50]	✓	
	PointConvFormer [124]	✓	

TABLE 2: Quantitative scene flow estimation results on FlyingThings3D [63] and KITTI [66] datasets. End-Point-Error (EPE) computes the mean Euclidean distance between the ground-truth and the scene flow prediction. Acc Strict calculates the percentage of points with EPE < 0.05m or relative error $< 5\%$; while Acc Relax calculates the percentage of points with EPE < 0.1m or relative error $< 10\%$. * indicates methods tested on datasets pre-processed by [31].

Methods		FlyingThings3D			KITTI		
		EPE (m)	Acc S. (%)	Acc R. (%)	EPE (m)	Acc S. (%)	Acc R. (%)
Feature Embedding	VoxFlowNet [74]	0.2971	11.36	33.46	-	-	-
	PV-RAFT* [114]	0.0461	81.68	95.74	0.056	82.26	93.72
	FlowNet3D [56]	0.1694	25.37	57.85	0.122	18.53	57.03
	FlowNet3D++ [112]	0.1369	30.33	63.43	0.253	-	-
	MeteorNet [57]	0.2090	-	52.12	0.2510	-	-
	FESTA [108]	0.1113	43.12	74.42	0.0936	44.85	83.35
	HPLFlowNet [31]	0.1318	32.78	63.22	0.119	30.83	64.76
	Just Go [71]	-	-	-	0.122	25.37	57.85
Cost-volume	PointPWCNet* [126]	0.0588	73.79	92.76	0.0694	72.81	88.84
	Res3DSF* [107]	0.0310	91.39	97.68	0.0351	89.32	96.20
Transformer	PT-FlowNet* [25]	0.0304	91.42	98.14	0.0224	95.51	98.38
	SCTN [50]	0.038	84.7	96.8	0.2549	23.79	49.57
	PointConvFormer [124]	0.0416	86.45	96.58	0.0479	86.59	93.32

and introduced two loss functions to train the network. One was the nearest neighbor loss which was able to push the combination of the first point cloud and the forward flow towards the next point cloud. Another one was the cycle consistency loss which forced the combination of the next point cloud and the reverse flow to be close to the first point cloud. With these simple loss functions design, they could finetune the network on other large SPL data no matter whether they had the ground truth annotations and achieved the state-of-the-art performance.

3.1.3 *Lattice-based methods*

Starting from PointNet [80] and PointNet++ [81], researchers always pre-process point clouds and chunk them into small blocks before sending the data into the network. In this way, the global information is inevitably damaged and leads to inaccurate boundaries as well. Lattice-based methods splat point clouds into lattice space which could further leverage the Bilateral Convolutional Layers (BCL) [41] to conduct scene flow feature learning. A typical lattice-based representation is shown in Figure 3c.

Inspired from the Bilateral Convolutional Layers (BCL) [41], HPLFlowNet [31] proposed a novel network which used the BCL and permutohedral lattice [2] to better estimate scene flow. The authors proposed DownBCL and UpBCL modified from the original BCL [41] to extract the lattice features and refine scene flow from the coarse estimation respectively. Moreover, a CorrBCL was introduced to better fuse the information from two separate and consecutive point cloud frames. HPLFlowNet also presented a new density normalization schema which made the network much more efficient and was able to generalize to various point densities.

3.2 Cost-Volume-based Methods

Cost-volume based methods for 3D scene flow estimation create a "cost" for potential motions of scene points. For each point, a volume of possible movements is predicted, and each movement has a cost based on how likely it fits observed changes across frames. The best movement for each point is the one with the lowest cost. This approach aims for smooth and consistent motion across the scene but can be computationally demanding due to the many potential motions considered.

PointPWC-Net [126] is the first work that exploring cost-volume based method to estimate scene flow in a coarse-to-fine manner inspired by FlowNet [39] and PWC-Net [98]. Specifically, to avoid the information loss in previous single flow embedding layer such as in FlowNet3D method [56], the authors built a pyramid network for point cloud and hierarchically refine scene flow. At each pyramid level, they warped the first point cloud features with the up-sampled coarse flow from the last level, and computed the cost volume with the second point cloud features. Finally, the refined scene flow was acquired after the scene flow

predictor. For supervised loss, they utilized the regular l2 loss for each layer between the groundtruth and the prediction. For the unsupervised loss, they introduced the Chamfer distance [21], smoothness constraint, and Laplacian regularization to predict the scene flow without any ground truth annotations.

Res3DSF [107] is developed, leveraging insights into human capabilities to discern dynamic movements in their environment. It integrates a context-aware module coupled with a residual flow refinement layer, all designed to achieve precise scene flow estimations. Several previous methodologies have often missed distinguishing repetitive patterns in dynamic environments. Res3DSF employs a distinct approach, incorporating contextual structure learning into its 3D spatial feature extraction layer and assimilating soft aggregation weights. A crucial aspect of this model is its optimization of attentive cost volume, which is pivotal for extracting flow embeddings from the context-enriched feature pyramid module. These embeddings subsequently undergo refinement via Three-NN interpolation and multiple MLP layers, culminating in the final thorough scene flow.

3.3 Transformer-based Methods

Transformer-based methods for 3D scene flow estimation employ the self-attention mechanism from transformers to capture intricate point-to-point relationships across consecutive frames. By processing both local and global context in point clouds, they ensure a comprehensive motion estimation. Adapting the transformer architecture, originally for text data, to the spatial nature of 3D scenes has proven to enhance the accuracy and consistency of motion predictions.

PT-FlowNet [25] is the first one introducing transformer architecture into the scene flow estimation task. It propose a novel approach employing point transformer (PT) extensively in its structure for optimal scene flow estimation in 3D environments. This unique integration of the transformer enables superior feature extraction from complicated point clouds. Additionally, the network utilizes a PT-based KNN branch within its iterative update module, allowing for more effective aggregation of correlated features compared to the conventional KNN with max-pooling. PT-FlowNet has exhibited exemplary performance and adaptability, especially on the FlyingThings3D and KITTI datasets, showcasing its effectiveness in real-world conditions.

SCTN [50] embraces an innovative voxel-based convolutional approach, ensuring coherent flows within three-dimensional spaces. It merges a sparse convolutional technique, aimed at profound feature extraction, with a transformer module to fortify the accuracy of scene flow predictions. This represents a pioneering integration of the transformer with sparse convolution, bestowing it with the capability to discern relational contextual information within point clouds. SCTN [50] calculates soft correspondences using a correlation matrix, integrating features extracted from both sparse convolution and the transformer module. To further amplify its discrimination of various motion fields, SCTN [50] introduces a feature-sensitive spatial consistency loss.

PointConvFormer [124] has re-engineered and refined the feature extraction mechanism through the use of transformers. This model has undertaken an in-depth exploration into the methodologies of calculating convolutional weights. Furthermore, PointConvFormer applies a Sigmoid activation function when dealing with attention weights, proving significantly more effective than the Softmax method. Owing to these insightful observations, PointConvFormer has manifested elevated performance in a series of trials compared to traditional Transformer models. Within the FlyingThings3D dataset, the EPE3D of PointConvFormer surpasses that of PointPWC-Net by 10%.

3.4 Discussion

The scene flow estimation results on both the synthetic dataset FlyingThings3D and real-world dataset KITTI are reported in Table 2. We have the following observations and discussions:

- Overall, the point-based and the lattice-based methods outperform the voxel-based methods by a large margin. This is because scene flow estimation is essentially a point-wise prediction task. The dense representation such as point and lattice are naturally fit with the task, while the voxel representation might suffer from losing the fine-grained information.
- Almost all types of methods demonstrate a well generalization ability from the synthetic domain to the real world domain. The models were trained on FlyingThings3D dataset and directly tested on KITTI dataset with promising performance. This reflects the potential of transfer learning and few-shot learning prospects on more real applications.
- Incorporating transformer models in 3D scene flow estimation can be highly beneficial. The Self-attention mechanism in transformers captures long-range dependencies and global interactions within scenes, enabling a more comprehensive understanding of scene dynamics. Multi-head self-attention provides multi-scale understanding, essential for capturing diverse scene features. Unlike conventional feature embedding or cost-volume, transformers allow efficient parallel processing, crucial for handling extensive 3D point cloud data, and accelerating training and inference. The flexibility of transformers enables integration with various architectures, enhancing feature capture. Their interpretability and representation learning capability make them a powerful tool for understanding intricate features and dynamic patterns within 3D scenes, offering a holistic and efficient approach to 3D scene flow estimation.

4 Point Cloud Detection

Object detection has been a significant computer vision task for a long time in both 2D and 3D domains which could bring tremendous applications such as self-driving, AR/VR, etc. The purpose is to recognize various objects and predict their precise bounding boxes in nature scenes. Previously, object detection in 2D images has made prominent achievements for both efficiency or accuracy. Meanwhile, motivated by the success of 2D object detection, research about 3D object detection is driving more and more attentions in the community. However, most of them still concentrate on using single-frame data as input. Recently, some researchers started to apply methods by taking multiple frames, which is SPL data, as the input for networks. Temporal information is investigated to obtain boosted detection results on 4D (3D spatial and 1D temporal) sequential data such as point clouds. Compared to object detection methods with only using single point cloud as input, sequential 4D data is more appealing, since it provides much richer context information and wide-range coverage of temporal consistency. The real world scenes are often dynamic and hard to predict. Objects might be missing or occluded between continuous frames. Leveraging spatio-temporal information can

TABLE 3: The summary of methods for multi-frame 3D object detection on sequential point cloud data.

Methods		Code	Attribute
Convolution-based	FaF [61]	✓	The Convolution-based methods learn temporal information by sliding window fusion schema which is convolution operation. This is more convenient but tends to lose details information or small objects.
	Second [130]	✓	
	IntentNet [12]	✓	
	What you see [34]	✓	
Graph-based	Yin et al. [135]	×	The Graph-based methods benefit from spatial features extracted from graph networks.
RNN-based	YOLO4D [19]	✓	Compared to simple convolution operation, the RNN-based methods aggregate temporal information better by exploring long-range temporal dependency. However, they usually cost more computation resources.
	McCrae et.al [64]	×	
	Huang et.al [37]	×	
Transformer-based	LIFT [137]	✓	Transformers are inherently suited to sequence-to-sequence tasks, allowing to effectively integrate temporal context, learning dependencies across different time frames.
	BEVFusion4D [10]	✓	
	FusionFormer [33]	✓	

significantly diminish false positives and false negatives during the object detection process. A list of multi-frame 3D object detection methods on SPL data is summarized in Table 3.

4.1 Convolution-based Methods

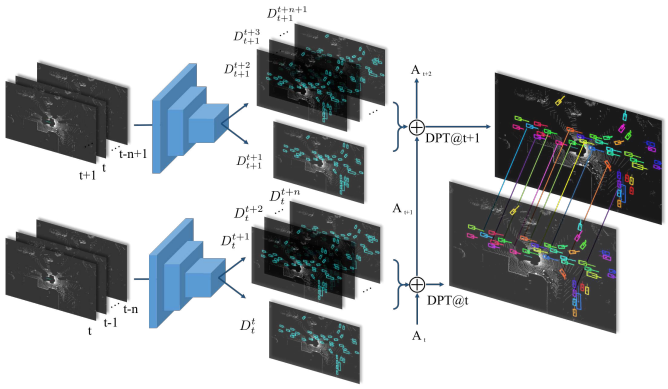


Fig. 4: The illustration of a convolution-based network for SPL object detection. The figure is from [61] with author's permission.

The convolution-based methods project SPL data into regular organization formats such as BEV (bird's eye view) map or voxel grid so that normal convolution operations could be leveraged to estimate object locations. A typical convolution-based network is shown in Figure 4. FaF [61] jointly conducted 4D object detection, tracking, and motion forecasting together which took full advantage of multiple point cloud frames as input. These sub-tasks were shown to associate each other and boosted up the performance. Each point cloud frame was represented by voxel. Nevertheless, FaF did not perform 3D convolution on 3D voxel due to the large computation cost. Instead, it operated 2D convolutions on the xy plane and directly treated the z dimension as feature information for 2D convolution. The same operation was applied for all of the frames and the coordinate system was normalized to be aligned across frames. The aggregated 4D tensor was sent to a single-stage object detector to accomplish the detection process. Meanwhile, to better utilize temporal information, FaF devised two schemes for temporal fusion. The early fusion directly concatenated tensors and used a 1D convolution to connect temporal features, while the late fusion hierarchically merged temporal features allowing the network to capture higher-level motion information. The object detection pipeline was the affinity of SSD [132] mentioned above. Tracking and motion forecasting will be introduced in Sec. 5 and Sec. 7.1.

Yan et al. [130] introduced an improved sparse convolution on voxelized point cloud leading to faster computation. Likewise, an angle loss function was added to deal with the limited object orientation prediction problem. The authors aggregated temporal information by concatenating multiple point cloud frames and considering time stamps information as additional features for network's input. IntentNet [12] proposed a fully convolutional network to deal with object detection and intent prediction at a single pass. It represented 3D point cloud from bird's eye view (BEV). The input data was modeled as 3D tensor and the height information was included as one of feature channels. Meanwhile, the temporal information from multiple Lidar sweeps was integrated into the height channel benefiting dynamic map and long trajectory predictions.

Hu et al. [34] argued that exploring free space for 2.5D data (RGBD or range image) is better than directly representing Lidar sweeps as 3D point clouds. The detection pipeline was built upon PointPillar [49] architecture. The visibility map was derived through raycasting algorithm from voxelized input data, which can be further blended into the network gradient learning process. During training, the visibility volume was treated as an additional input to the network by two fusion methods, early fusion, and late fusion. The difference between these two fusion methods is located whether to compute input features separately using the backbone network. The aggregation of temporal information was considered to be an augmenting trick by taking the advantage of visibility prior. The authors of [34] compensated motion by transferring SPL into a single scene and encoding timestamps as an additional input along with xyz geometry, which can be proven to improve detection results by a large margin over PointPillar [49] baseline model.

The essence of 4D-Net [77] is its pioneering dynamic connection learning, rooted in a meticulous convolution process. This method is designed to enable an advanced fusion of varied feature representations from diverse modalities and abstraction levels, all while rigorously preserving geometric fidelity. Through dedicated convolutional architectures, each modality yields a plethora of rich features that are strategically aligned and integrated, facilitating a seamless interaction and synthesis of 4D information from assorted sensors. Unlike preceding models, 4D-Net initiates the convolution early in the workflow, mitigating the dilution of vital spatial data and optimizing the use of motion cues and high-density image information. This intricate convolution-driven approach substantially augments the detection proficiency in multifaceted spatial and temporal environments.

TABLE 4: Quantitative results of 3D object detection on Waymo Open Dataset [99] *val* set (vehicles and pedestrians).

Method		Vehicles				Pedestrians			
		3D AP		BEV AP		3D AP		BEV AP	
		IoU=0.7	IoU=0.8	IoU=0.7	IoU=0.8	IoU=0.5	IoU=0.6	IoU=0.5	IoU=0.6
Single-frame Methods	StarNet [72]	53.70	-	-	-	66.80	-	-	-
	PointPillar [49]	60.25	27.67	78.14	63.79	60.11	40.35	65.42	51.71
	MVF [144]	62.93	-	80.40	-	65.33	-	74.38	-
	AFDET [27]	63.69	-	-	-	-	-	-	-
	RCD [8]	68.95	-	82.09	-	-	-	-	-
	PillarNet [110]	69.80	-	87.11	-	72.51	-	78.53	-
	PV-RCNN [94]	70.47	39.16	83.43	69.52	65.34	45.12	70.35	56.63
	MVF++ [78]	74.64	43.30	87.59	75.30	78.01	56.02	83.31	68.04
Multi-frames Methods	Huang et al. [37]	63.60	-	-	-	-	-	-	-
	MVF++ [78]	79.73	49.43	91.93	80.33	81.83	60.56	85.90	73.00
	LIFT [137]	69.0	64.2	-	-	69.9	65.3	-	-
	FusionFormer [33]	79.73	49.43	91.93	80.33	81.83	60.56	85.90	73.00
	Qi et al. [78]	**84.50**	**57.82**	**93.30**	**84.88**	**82.88**	**63.69**	**86.32**	**75.60**

TABLE 5: Quantitative results of 3D object detection on nuScenes [9] dataset. "T.C." stands for traffic cone. "Moto." and "Cons." represent motorcycle and construction vehicle, respectively.

Method		Car	Pedestrian	Bus	Barrier	T.C.	Truck	Trailer	Moto.	Cons.	Bicycle	Mean
Single-frame Methods	VIPL_ICT [73]	71.9	57.0	34.1	38.0	27.3	20.6	26.9	20.4	3.3	0.0	29.9
	MAIR [96]	47.8	37.0	18.8	51.1	48.7	22.0	17.6	29.0	7.4	**24.5**	30.4
	PointPillars [49]	68.4	59.7	28.2	38.9	30.8	23.0	23.4	27.4	4.1	1.1	30.5
	SARPNET [134]	59.9	69.4	19.4	38.3	44.6	18.7	18.0	29.8	11.6	{14.2}	32.4
	Tolist [73]	79.4	71.2	42.0	**51.2**	47.8	**34.5**	34.8	36.8	9.8	12.3	42.0
Multi-frames Methods	FusionFormer [33]	-	-	-	-	-	-	-	-	-	-	72.6
	What you see [34]	79.1	65.0	46.6	34.7	28.8	30.4	40.1	18.2	7.1	0.1	35.0
	McCrae et al. [64]	67.97	56.87	-	-	-	-	-	-	-	-	-
	Yin et al. [135]	79.7	76.5	47.1	48.8	58.8	33.6	43.0	40.7	18.1	7.9	45.4
	LIFT [137]	87.7	86.1	62.4	69.3	83.2	55.1	59.3	70.8	29.4	47.7	65.1
	BEVFusion4D [10]	89.7	90.9	72.9	81.0	87.7	65.6	66.0	79.5	41.1	58.6	73.3

4.2 RNN-based Methods

These methods [19], [37], [64] investigated recurrent neural networks to capture the temporal consistency of detection features and improved object localization accuracy. Figure 5 depicts a general idea of the RNN-based methods. The network extracts spatial features by CNN for each point cloud frame. Then a recurrent network dubbed ConvLSTM is integrated to learn temporal features from previous state and current state, leading to generated features for the next layer. The paper [37] proposed by Huang et al. was the first one that modeled temporal relations among SPL with an RNN-based (LSTM) schema to boost up the performance of 3D/4D object detection results. The proposed network took SPL as input and generated backbone features for each point cloud frame by a 3D Sparse Conv U-Net. A novel 3D sparse LSTM was used to fuse backbone features across previous timestamp $t-1$ and current timestamp t. After embedding temporal information into hidden features, object proposals for each point were predicted by an object detection head network. Moreover, the authors built a knowledge graph among all of the point nodes to enhance spatial geometry information and suppress false positives. The final object detection results were refined by a traditional non-maximum suppression algorithm.

Besides simple stacking LSTM layers which just concatenated SPL frames as 4D tensor and used CNN to comprehend temporal information, another way is to adopt powerful ConvLSTM. YOLO4D [19] extended YOLO v2 [84] network to 3D space and leveraged not only spatial but also temporal information from SPL. It could capture temporal information better and exhibited

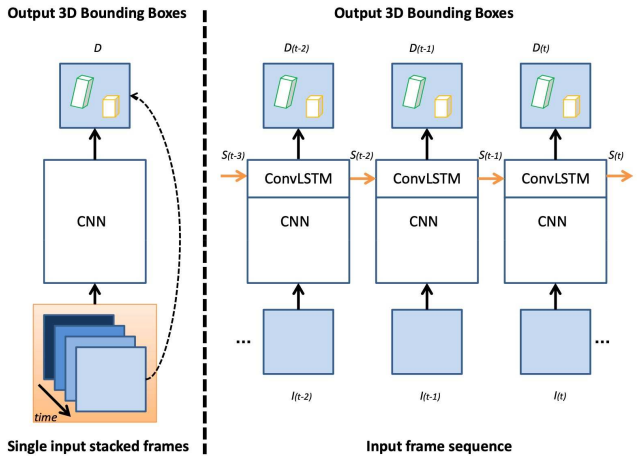


Fig. 5: The illustration of an RNN-based method for SPL objection detection. The figure is from [19] with author's permission

superiority during the multiple frames object detection process. McCrae et al. [64] employed PointPillar [49] as its baseline and developed a recurrent designed network that specifically takes three point cloud frames as input. Each point cloud frame was processed by a PointPillar model to extract features and followed a ConvLSTM to model temporal relation between the past and current time stamps. These designs were shown to be effective in pedestrian and vehicle classes.

4.3 Graph-based Methods

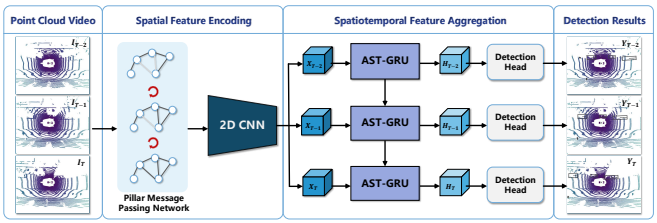


Fig. 6: The illustration of a graph-based method to conduct object detection. The figure is from [135] with author's permission.

The core idea of these methods is to explicitly capture point spatio-temporal strictures with graph networks modeling. Figure 6 demonstrates a graph-based network to generate detection results. The network took SPL as input and all of the frames were aligned to the same coordinate system to eliminate ego-motion effects. After spatial features were extracted from point cloud frames, they were sent to Attentive Spatiotemporal Transformer Gated Recurrent Unit (AST-GRU) network to perform temporal information accumulation which can aid dynamic object detection results. Yin et al. [135] explicitly proposed an object detection method from sequential point clouds and explored the superiority over single-frame 3D object detection which has limitations of sparse, occlusion and bias sampling, etc. A delicate PMPNet was developed to manipulate the spatial relation from the encoded pillar grids graph in an iterative message-passing manner.

4.4 Transformer-based Methods

Transformer-based methods for 4D object detection merge the strengths of transformers in handling long-range dependencies with the challenges of detecting objects in 3D space over time, offering promise for more robust detection and tracking of objects in dynamic scenes.

The LIFT [137] (LiDAR Image Fusion Transformer) method employs 4D sequential cross-sensor data alignment to assimilate temporal interactions between LiDAR and camera sensors over successive time frames. Specifically, LIFT uses transformer architectures, enabling the model to aggregate multi-frame, multi-modal information over time, accentuating temporal variations. By utilizing bird-eye-view projections and computing sparse grid-wise self-attention, LIFT maintains temporal coherence with reduced computational load, delivering enhanced 3D object detection in dynamic autonomous driving scenarios, as validated on the nuScenes and Waymo datasets.

BEVFusion4D [10] stands as an advanced fusion framework for 3D object detection in autonomous driving, integrating LiDAR and camera information into a Bird's-Eye-View (BEV) using a transformative approach. A pivotal component is the LiDAR-Guided View Transformer (LGVT), which acts as a sophisticated transformer model, utilizing LiDAR-derived spatial priors to optimize the extraction of relevant semantic information from camera views in the BEV space effectively. Furthermore, the framework incorporates a Temporal Deformable Alignment (TDA) module, employing transformer methodologies to aggregate historical frame features, thereby providing a comprehensive spatiotemporal representation. This transformative approach significantly elevates BEVFusion4D's performance, rendering it superior on the nuScenes datasets with a leading edge in spatial and spatiotemporal detection scenarios.

FusionFormer [33] is a pioneering end-to-end framework devised for refined 3D object detection, leveraging transformers to facilitate precise multi-modal fusion, addressing the Z-axis information loss seen in conventional methods. This framework permits features to be inputted in their original forms and utilizes deformable attention to integrate LiDAR and image features effectively. FusionFormer introduces a specialized depth prediction branch, optimizing camera-based detection tasks, and a novel plug-and-play temporal fusion module, utilizing deformable attention for the assimilation of historical BEV features, yielding enhanced detection stability and reliability.

4.5 Discussion

4D SPL object detection results on benchmarks of Waymo Open and nuScenes Datasets are summarized in Tables 4 and 5, respectively. Here are the observations and discussions:

- On both benchmarks of Waymo Open and nuScenes Datasets, the multi-frame methods demonstrate a clear superior performance compared to the single-frame methods. Although more information is involved, this does reflect the essence of additional temporal information. By using SPL data and devising spatio-temporal feature extracting techniques to conduct object detection, those false bounding box results are largely suppressed to ensure temporal consistency and thus improve overall detection accuracy.
- Compared to the RNN-based methods, the convolution-based and the graph-based methods accomplish better performance on nuScenes benchmark. As we also discussed in Supplementary, the RNN-based networks exploit more on temporal relations among long-range time series, while high-level semantic understanding tasks like detection prefer temporal consistency in both spatial and temporal domains.
- Almost all of the multi-frame detection methods are restricted to less than 10 frames. Thus long-range SPL object detection still remains as a challenging problem.
- Besides the above mentioned methods, Qi et al. [78] explored an offboard application yielding groundtruth 3D labels by utilizing SPL detection results which have sufficient context information. The authors followed the similar method of [34] which aggregated temporal information by transforming other point cloud frames to the current one to get rid of ego-motion and encoded time offsets as an additional feature. Meanwhile, it reached the state-of-the-art 3D object detection performance on challenging Waymo Open Dataset.

5 Point Cloud Tracking

4D multi-object tracking (MOT) is another essential application of SPL, which is also a vital component for autonomous driving task cooperated with 4D object detection prior. Being aware of object locations in each point cloud frame, 4D MOT takes the responsibility of associating them together in a whole sequence. The temporal consistency plays a crucial role to cope with the tracking problem in this process. Normally, 4D MOT system follows 2D MOT schema while the difference is the detection process happens in 3D space. In recent years researchers start to directly utilize 3D point cloud data to perform MOT even without any additional features such as RGB information.

TABLE 6: The summary of the Multiple Object Tracking methods.

Methods		Code	Attribute
Filter-based	AB3DMOT [117]	✓	The 3D based methods are more easy to implement and get rid of relying on other data modalities. However, this usually are less sensitive to the extreme motion.
	Complexer-YOLO [95]	✓	
	Chiu et al. [15]	✓	
	Giancola et al. [28]	✓	
	DSM [24]	✓	
Temporal Convolution	P2B [82]	✓	The joint 2D&3D-based methods could be associative with the detection pipeline and are usually more accurate due to additional semantic signals from the 2D RGB modality. However, the large computation cost is also inevitable.
	FaF [61]	✓	
	PointTrackNet [109]	✓	
	GNN3DMOT [120]	✓	
	mmMOT [139]	✓	

TABLE 7: Quantitative 3D MOT Results of on KITTI Test Dataset.

Method		MOTA↑	MOTP↑	MT↑	ML↓	ID_sw↓	FRAG↓
Filter-based	Complexer-YOLO [95]	75.70	78.46	58.00	5.08	1186	2092
	AB3DMOT [117]	83.84	85.24	66.92	11.38	**9**	**224**
	Chiu et al. [15]	-	-	-	-	-	-
	Giancola et al. [28]	-	-	-	-	-	-
	DSM [24]	76.15	83.42	60.00	8.31	296	868
Temporal Convolution	FaF [61]	80.9	**85.3**	55.4	20.8	-	-
	PointTrackNet [109]	68.23	76.57	60.62	12.31	**111**	725
	GNN3DMOT [120]	80.40	85.05	70.77	11.08	113	**265**
	mmMOT [139]	**84.77**	**85.21**	73.23	**2.77**	284	753

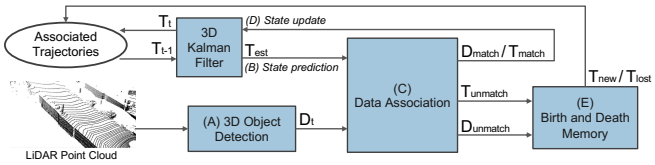


Fig. 7: A baseline for 3D-based MOT methods. The figure is from [117] with author's permission.

5.1 Filter-based Methods

Recent methods [15], [28], [61], [109], [117] operate only raw SPL data for 3D MOT task. Usually these methods rely on object detectors to provide object locations and various filter-based algorithms to predict object trajectories, as shown in Figure 7. AB3DMOT [117] provided a compact baseline for multi-object tracking task while maintaining high-efficiency meeting the real-time estimation requirement. In this work, the authors derived detection results for the current Lidar frame through a pre-trained 3D object detector. The 3D Kalman Filter with constant velocity model predicted the state of object trajectory from previous frame. The predicted trajectory and detected objects were associated with Hungarian algorithm in current frame, which can further update trajectory state in 3D Kalman Filter. The authors also regularized the evaluation of 4D MOT system directly in 3D space instead of projecting into 2D plane as the previous work did. A new evaluation tool and three evaluation metrics were proposed to evaluate tracking performance on self-driving benchmarks in a more reasonable manner. Similar to the paper proposed by Weng et al. [117], Chiu et al. [15] dealt with the tracking problem using 3D Kalman Filter with a constant linear and angular velocity model. Besides the traditional approach, the authors exploited Mahalanobis distance for data association process and co-variance matrices for the state prediction process.

In addition to SPL input, there are methods involving another modality RGB image to the network as well. The features from different domains could complement each other and lean-to more representatives. DSM [24] was an earlier work leveraging the deep structured model to create multiple neural networks together to solve the 4D MOT task. It predicted object proposals using a Detection Network from the input point cloud and RGB sequence. After formulating discrete trajectories, a liner optimization process was utilized to generate final tracking results. To utilize high-level semantic features for 3D MOT task, the authors of [95] generated semantic segmentation maps from input images. The semantic information was further back-projected to 3D space to obtain class-aware point clouds and provide extra semantic guidance to the tracking process. They predicted 3D bounding boxes from the voxelized semantic point cloud. The Scale-Rotation-Translation score (SRTs) was devised to reasonably evaluate performance and accelerate the speed to real-time.

5.2 Temporal Convolution Methods

However, previous filter-based methods were not sensitive to the extreme motion condition which may harm tracking performance. PointTrackNet [109] designed PointTrackNet to conduct object detection first from two continuous point cloud frames. The locations were further refined by an association model to merge detection results and ameliorate the impact of the false positive. The final tracklets can be provoked by linking matched objects. P2B [82] coped with the tracking problem with a point-wise schema and without using a traditional Kalman filter which has a relatively large computation cost. It proposed an end-to-end network and treated the tracking task as the detection task inspired by VoteNet [79]. The sampled seeds and target centers embedded with local geometry information were first extracted from sequential point clouds. This strengthened the object representation instead of using single bounding box such as [28]. Then each target center was clustered with its neighbors to form the target proposal. Finally, object proposals were further verified over the whole sequence to ensure 3D appearance consistency and acquire tracking results.

Inspired by paper [1], to promote tracking performance with both richer feature representations and the regularization of the shape completion, Giancola et al. [28] proposed the first 4D MOT

Siamese network structure. Specifically, first, the features extracted by an encoder network served as compact latent representations for Siamese tracker. Then the cosine similarity metric was used to match candidate shapes with model shapes. Finally, the decoder part of the shape-completion network was added to regularize Siamese tracker which could ensure the meaningful latent representation.

Different from the normal trajectory optimization solution, FaF [61] solved the tracking problem in an associative manner, incorporating with the object detection, motion forecasting and tracking tasks into a single pipeline. Firstly, as mentioned in Sec. 4, FaF adopted multi-frame object detection methods to derive object bounding boxes locations for the whole sequential frames. A motion forecasting algorithm was applied to predict object locations in further time stamps. In conjunction with past and current locations, tracklets could be obtained through average fusion.

Unlike previous work such as [117] extracting object features independently to perform the Hungarian data association, GNN3DMOT [120] offered a novel multi-modality feature extractor to learn motion and appearance features from both 2D and 3D spaces. Furthermore, they firstly introduced a graph-based pipeline exploring the feature interaction among various objects to derive a more discriminate affinity matrix. Consequently, the data association process could benefit a lot from valuable object features which could also lead to a boosted tracking performance.

5.3 Discussion

Table 7 summarizes 4D multi-object tracking (MOT) results on the KITTI benchmark. Several observations and discussions are listed below:

- Compared to pure 3D-based methods, joint 2D&3D-based methods are more frequently used by the recent research community with a relatively higher performance, which shows the superiority of more modalities.
- Most high-performance methods still require an additional 2D input to ensure tracking accuracy. This is a limitation with extra data. In the real self-driving scenario, usually, it costs much more to process multi-modalities at the same time.
- For almost all of 3D MOT methods, tracking performance is based on detection performance. Only PointTrackNet [109] and P2B [82] belong to a full end-to-end pipeline breaking the limit of the off-shell detector. However, their performance is not satisfied which leaves a potential improvement for future research on this track.
- Compared to MOT, 4D Single Object Tracking (SOT) aimed to estimate the object state in further frames based on the previous state. Pang et al. [75] recently investigated 4D Single Object Tracking (SOT) and obtained tracklets through estimated object bounding boxes at various time stamps. The tracking process can be treated as a multi-frames registration method.

6 4D Point Cloud Segmentation

Segmentation has always been another prevalent and crucial topic for high-level scene understanding including semantic segmentation, instance segmentation, and the combined version, panoptic segmentation. Distinct from detection and tracking, segmentation tasks demand a more fine-grained understanding of the surrounding scene. They require a pixel or point level classification for diverse scene object categories which could also provide a more holistic perception. Based upon previously developed 2D or single frame 3D segmentation methods, 4D segmentation over SPL recently gains amounts of popularity due to real applications in our dynamic world such as AR/VR, self-driving, etc. The path of handling the extra temporal dimension and keeping consistency in the 4D spatio-temporal space is paved by community.

A list of SPL segmentation methods is summarized in Table 8. In the following sections, we will cover 4D point cloud semantic and panoptic segmentation in Sec. 6.1 and Sec. 6.2 respectively.

6.1 4D Semantic Segmentation

The purpose of semantic segmentation is to apprehend semantic information from surrounding scenes and forecast the class label for each point in the point cloud. However, information provided by a single frame is usually limited. To get a relatively comprehensive perception of the real world, it is indispensable to explore approaches of fusing temporal information across multiple frames.

6.1.1 Simple Gathering

Some methods claim that the 4D semantic segmentation task can be simplified into the related 3D one. Given SPL which have multiple frames, a network gathers point clouds into a single frame by transferring other frames' data into the coordinate system of the current frame. Then 3D semantic segmentation methods can be applied to solve the problem.

Projection-based One large category of 3D semantic segmentation methods is the project-based methods. The input point clouds are primarily projected to the BEV (Bird's Eye View) or the spherical space and then 2D segmentation pipelines can be easily applied to 2D projected data. Taking the advantage of advanced 2D CNN networks, the 3D segmentation process can be significantly sped up. Zhang et al. [138] and PolarNet [140] followed the BEV (Bird's Eye View) projection track which format scene with a top-down snapshot. The network output segmentation results on the 2D spatial location including the semantic class prediction of the voxel along the Z-axis. Although these BEV project methods accomplished promising performance on segmentation benchmarks, scene information loss was inevitable. Spherical projection aimed to project point cloud data into the 360° spherical space and then flatten it to the 2D image which can maintain maximum information. The resulted spherical projection image indicated structural information from the camera viewpoint. Studies [70], [121], [122], [129] followed the spherical projection track which treated the range image as the input data representation and predicted segmentation results with 2D CNN networks. In conjunction with some post-filtering technologies, 3D point cloud could be reconstructed from the range image.

Convolution-based Researchers also represented 3D point cloud data with regular grids so that 3D convolution operations could be applied to learn semantic features. Some studies [36], [60], [65], [85], [102] transferred point cloud to voxel representation and adopted 3D convolutions over 3D volume data to estimate segmentation results for each occupancy grid. Although it was more straightforward to perform 3D semantic segmentation, 3D voxel convolution still suffered from the heavy computation cost and representation redundancy, leading to the inevitable accuracy and efficiency loss. Papers [88], [97] splatted point cloud into the permutohedral lattice space to perform sparse convolutions. The lattice representation enables convolution operations to learn the semantic segmentation prediction while preserving maximum

information at the same time. Octree [90] was another approach to formatting point cloud data. Octnet [86] was devised to conduct convolution operations on the octree structure for point cloud. PointConv [125] extended the convolution on the 2D image to the 3D domain with the dynamic filter which supported both the convolution and deconvolution. KPConv [103] proposed Kernel Point deformable Convolution to cope with more flexible point cloud.

Point-based Likewise, there is still another popular category directly processing 3D point clouds to estimate semantic segmentation results. Pioneered by PointNet [80], the authors proposed a shared-MLP based network and output point-wise labels for each point. Due to the lack of enough local geometry information, PointNet++ [81] attempted to add the grouping operation at multiple scales and resolutions to grab both local and global semantic features. Inspired by PointNet and PointNet++, a tremendous of point-based methods such as [20], [35], [43], [141], [143] have been investigated to estimate semantic scene labels for point clouds. They exploited all kinds of different ways to aggregate representative features from local neighbors and promote segmentation performance. Some other methods such as [14], [104], [131], [142] introduced the attention mechanism to point-based networks to help extract more critical points and benefit segmentation results.

6.1.2 Temporal Convolution

Simply gathering multiple frames into a single channel inevitably losses much spatial and temporal information especially when there are large motions or deformations between frames. Instead of simply gathering, studies explored more advanced approaches [16], [18], [22], [57], [93] to learn the temporal information for the 4D semantic segmentation on sequential point clouds.

Grid-based Convolution These methods [16], [93] transferred point clouds to the regular data representation such as voxel occupancy and convolution operations could be applied along both spatial and temporal dimensions. Thus, the high-level context information could be fused across multiple frames and better inferring semantic perception in each frame. To achieve the point-wise semantic label prediction purpose, 4D MinkNet [16] was the first method that applied the deep convolution network on high dimensional data such as SPL. It adopted the idea from Sparse Tensor [29] and proposed the generalized sparse convolution to operate high dimensional data. The proposed convolution layer can be integrated with various deep networks and well generalized to different tasks. To deal with the computational problem when generalizing convolution to high dimensional spaces, the authors designed a novel kernel that is not hyper cubic and thus reduces the memory cost. The 4D segmentation network inherited the traditional 2D segmentation design U-Net [87] including sparse convolutions and sparse transpose convolutions. The skip connection was also adopted to link low-level and high-level layers.

Although U-Net is a conventional method for semantic segmentation problem, its basic structure could still fail in some complex and dynamic scenarios. To better fuse global and local features, SpSequenceNet [93] leveraged two novel models upon U-Net baseline to improve the segmentation performance, the Cross-frame Global Attention (CGA) and cross-frame local interpolation (CLI). The entire network structure took two consecutive frames as input and followed the U-net design in paper SSCN [30] which contained 3D residual blocks in the encoder part. The Cross-frame Global Attention (CGA) model was utilized to import global attention information. It generated a mask from the previous frame which contained crucial semantic features such as appearance information. The mask could further guided the current frame feature extraction. Another model cross-frame local interpolation (CLI) was inspired by the scene flow embedding layer and fused both spatial and temporal feature information.

Point-based Convolution While grid-based methods are relatively consistent with 2D segmentation pipeline, they still suffer from quantization errors which lose information ineluctably. Compared to them, point-based convolution networks [22], [57] are usually more compact. They capture features from raw SPL data which preserve most object details information. MeteorNet [57] directly processed raw SPL data and performed spatio-temporal feature learning using a similar structure as PointNet++ [81] which has been introduced in Supplementary. As for 4D semantic segmentation networks, MeteorNet built MeteorNet-Seg to conduct point-wise semantic label prediction process. The MeteorNet-Seg harnessed the Meteor-ind [57] module and the early-fusion strategy to construct the network. The Meteor-ind [57] module only contained neighbor points for each local patch due to point correspondence was not important for the segmentation task. The early-fusion strategy combined input point clouds early before the network to fuse temporal information.

PSTNet [22] was another concurrent work designed for processing SPL with spatial-temporal convolution. The authors devised a Point tube structure to organize input data more efficiently and conduct proposed PST convolution. The point tube incorporated spatial and temporal kernels separately to capture spatio-temporal local structure information. To perform the point-level prediction task such as 4D semantic segmentation, the PST transposed convolution was developed to recover spatial and temporal scales which had been down-sampled by the PST convolution. Overall a hierarchical structure was built to process spatial and temporal features at different levels for 4D semantic segmentation task. Compared to grid-based methods such as [16], PSTNet was more compact yet effective while 4D MinkNet [16] has a relatively large representation redundancy, especially with an increasing scale of data.

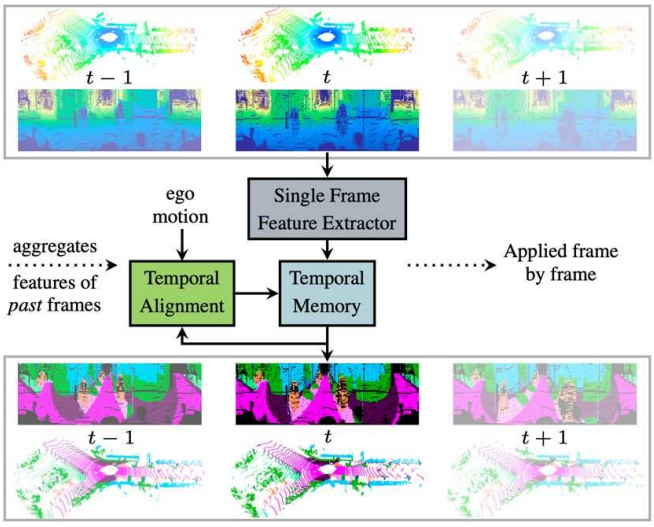


Fig. 8: The illustration of a RNN-based method for 4D semantic segmentation. The figure is from [18] with author's permission.

TABLE 8: The summary of the sequential point cloud segmentation methods.

Methods			Code	Attribute
Semantic Segmentation	Grid-based Convolution	MinkNet [16]	✓	The grid-based convolution methods are more convenient to implement due to the regular gird representation of the point clouds, while inevitably suffer from the quantization error.
		SpSequenceNet [93]	✓	
	Point-based Convolution	MeteorNet [57]	✓	Point-based convolution preserve more information from the raw point clouds.
		PSTNet [22]	✓	
	RNN-based Convolution	Duerr et al. [18]	×	Explicitly learn the temporal information but with higher computation cost.
Panoptic Segmentation	Point-based	Aygün et al. [4]	✓	Jointly learning mutually boost each other and get a more holistic scene understanding.
		PanopticTrackNet [38]	✓	

TABLE 9: Quantitative semantic segmentation results on SemanticKITTI multiple scans dataset (IoU (%)). The ⋆ shows moving classes.

Methods	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic sign	car⋆	bicyclist⋆	person⋆	motorcyclist⋆	other-vehicle⋆	truck⋆
TangentConv [101]	34.1	84.9	2.0	18.2	21.1	18.5	1.6	0.0	0.0	83.9	38.3	64.0	15.3	85.8	49.1	79.5	43.2	56.7	36.4	31.2	40.3	1.1	6.4	1.9	**30.1**	**42.2**
DarkNet53Seg [6]	41.6	84.1	30.4	32.9	20.2	20.7	7.5	0.0	0.0	91.6	**64.9**	75.3	**27.5**	85.2	56.5	78.4	50.7	64.8	38.1	53.3	61.5	14.1	15.2	0.2	28.9	37.8
SpSequenceNet [93]	43.1	88.5	24.0	26.2	29.2	22.7	6.3	0.0	0.0	90.1	57.6	73.9	27.1	**91.2**	**66.8**	**84.0**	**66.0**	**65.7**	50.8	48.7	53.2	41.2	26.2	**36.2**	2.3	0.1
Duerr et al. [18]	**47.0**	**92.1**	**47.7**	**40.9**	**39.2**	**35.0**	**14.4**	0.0	0.0	**91.8**	59.6	**75.8**	23.2	89.8	63.8	82.3	62.5	64.7	**52.6**	**60.4**	**68.2**	**42.8**	**40.4**	12.9	12.4	2.1

TABLE 10: Quantitative semantic segmentation results on the Synthia 4D dataset.

Methods		Input	#Params (M)	mIoU (%)
Single Frame	3D MinkNet14 [16]	voxel	19.31	76.24
	PointNet++ [81]	point	0.88	79.35
Multiple Frames	4D MinkNet14 [16]	voxel	23.72	77.46
	MeteorNet [57]	point	1.78	81.8
	PSTNet (l = 1) [22]	point	1.42	80.79
	PSTNet (l = 3) [22]	point	1.67	**82.24**

TABLE 11: Quantitative 4D panoptic segmentation on SemanticKITTI validation set. MOT (Multiple Object Tracking) method by [115]; SFP (Scene Flow Propagation) Method by [71].

Method		LSTQ	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
MOT	RangeNet++ [70]	24.06	52.43	64.52	35.82	42.17
	KPConv [103]	25.86	55.86	66.90	47.66	54.13
	Aygün et al. [4]	40.18	28.07	57.51	66.95	51.50
SFP	RangeNet++ [70]	34.91	23.25	52.43	64.52	35.82
	KPConv [103]	38.53	26.58	55.86	66.90	47.66
	Aygün et al. [4] (1 scan)	43.88	33.48	57.51	66.95	51.50
	Aygün et al. [4] (4 scans)	56.89	56.36	57.43	66.86	51.64

RNN-based Convolution The RNN-based Convolution methods choose to aggregate temporal information recurrently as shown in Figure 8. Specifically, for each time stamp t, the network fused information from the previous frame at time $t-1$ and strengthened the segmentation of the current frame. The feature of the current frame would be continued to enhance future frames. Duerr et al. [18] projected each point cloud in a sequence to the image plane dubbed as range image mentioned in Sec. 6.1.1 and input to the network. For the entire sequence, the semantic feature would be perpetually reused instead of used just once in the previous paper such as SpSequenceNet [93]. During temporal memory update process, the authors utilized two recurrent strategies to perform the feature fusion. One was adopting Residual Network which concatenates the past frame feature information with the current one and used MLP layers to conduct the spatial fusion. Another was ConvGRU dubbed as Gated Recurrent Unit which introduced gating mechanisms and replaced the MLP layer with the convolution layer. The latter one was a better choice which was able to achieve trade-off between efficiency and efficacy.

6.2 4D Panoptic Segmentation

Panoptic segmentation is a merged joint segmentation task including semantic segmentation and instance segmentation, which was first introduced in [48] in the image space and further extended from image to video by [47]. Behley et al. [7] presented a large-scale Lidar benchmark for point cloud panoptic segmentation, in conjunction with baseline results for single-scan segmentation performance. Inspired from image to video upgrading in the 2D space and also the existing single-scan point cloud panoptic segmentation baseline, Aygün et al. [4] firstly proposed a 4D Panoptic Segmentation pipeline demonstrated in Figure 9. The authors took a sequence of point clouds as input and inferred semantic classes for each point along with identifying the instance ID, completing both semantic and instance segmentation jointly for SPL. They first clustered points anchored on object center seeds and then assigned semantic information for each point. One major contribution in the paper was standardizing the evaluation protocol for the sequentially panoptic segmentation problem by devising a new point-centric evaluation method. Compared to existing metrics PQ [48] and MOTA [105] which had problems of over-estimating small segments and under-estimating frame association separately, the proposed LSTQ (LiDAR Segmentation and Tracking Quality) unified the evaluation in space and time domains and measured point-to-instance association quality.

To explore a holistic scene understanding problem, PanopticTrackNet [38] blended panoptic segmentation and multi-object tracking tasks. It proposed a novel architecture PanopticTrackNet with post-processing which unified semantic segmentation, instance segmentation, and multi-object tracking. The PanopticTrackNet was a multi-head end-to-end network containing a semantic segmentation head, instance segmentation head, and instance tracking head which simply concatenated frame vectors to merge temporal information. It took continuous RGB frames or point clouds as input and generated segmentation results. Then the MOPT fusion model was applied to predict the pixel-wise panoptic tracking output.

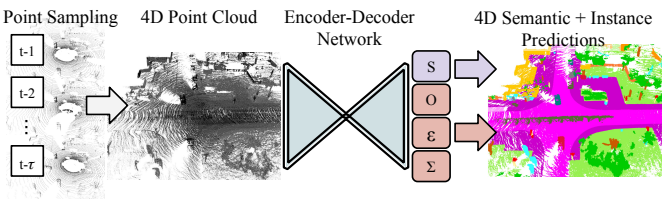


Fig. 9: The illustration of a typical 4D panoptic segmentation method. The figure is from [4] with author's permission.

6.3 Discussion

We summarize semantic segmentation results on the SemanticKITTI multiple scans benchmark and Synthia 4D dataset in Tables 9 and 10, respectively. The results of 4D Panoptic Segmentation on SemanticKITTI [6] dataset are reported in Table 11. Based on these tables, we have the following observations and discussions:

- Additional temporal data improves the overall segmentation accuracy by a large margin compared to static point cloud methods as shown in Table 9, especially for those moving object classes. The motion information is well-captured by 4D semantic segmentation methods which further enhance the temporal consistency and remove false segmentation results.
- From Table 10, point-based convolution outperforms grid-based convolution in terms of both efficacy and efficiency. Especially for efficiency, the number of parameters of point-based is much less than the grid-based methods, which avoids large computation cost of the quantization process.
- Overall segmentation performance is still limited on moving object classes which shows the large impact of motion information.
- The panoptic segmentation methods significantly outperform other basic segmentation methods by exploring a holistic semantic scene understanding. The increase of scan numbers brings consistent performance gain.

7 Point Cloud Forecasting

Besides getting the perception of the surrounding world such as detection and segmentation, future forecasting is another critical component for a more holistic scene understanding. The reasonable and precise future prediction would largely decrease the uncertainty during motion planning or self-driving process, especially in 3D space. Point cloud forecasting takes previous history information into the system and generates future object positions or entire scene point clouds, which would classify the task as motion forecasting or sequential forecasting. A list of point cloud forecasting methods is summarized in Table 12.

In the following sections, the motion forecasting will be presented in Sec. 7.1 and the sequential forecasting will be summarized in Sec. 7.2.

7.1 Point Cloud Motion Forecasting

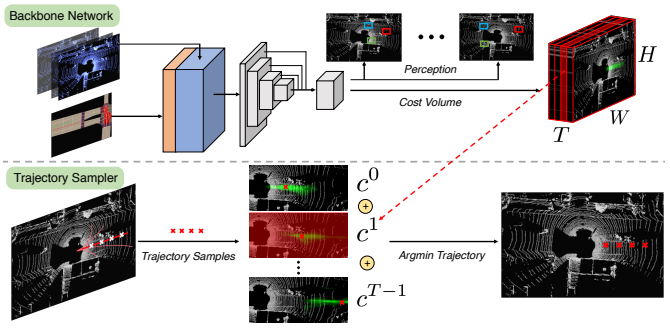


Fig. 10: The illustration of a voxel representation method for motion forecasting. The figure is from [136] with author's permission.

Motion forecasting, also called motion prediction, aims to predict future object positions and trajectories by accumulating history spatial-temporal information. The conventional solution to this problem is usually associated with object detection and tracking, since knowing past object locations would provide strong prior knowledge to the future prediction. Usually, these methods are applied to image sequences or video signals by availing of powerful CNN networks. While high demands arise for predicting the future from raw sensor data, the community starts to explore motion forecasting from point clouds [11], [12], [61], [69], [91], [123], [136].

7.1.1 BEV Representation

Since point cloud data are usually sparse and irregular, one convenient and efficient way is adopting the Bird's Eye View (BEV) representation, which converts point clouds to 3D tensors [11], [12], [61], [136]. Besides the XY location, height is treated as another feature to form one channel. In this way, clear separations between target objects could still be preserved while largely reducing computation cost for high-dimensional data. Figure 10 shows a typical BEV representation method for motion prediction.

As introduced in Sec. 4 and Sec. 5, FaF [61] was also the first one proposing a holistic network that jointly conducted object detection, tracking and motion forecasting from SPL input. Due to the association among multiple tasks, FaF had attained good fidelity for the motion prediction by adopting BEV representation. The IntentNet [12] (introduced in Sec. 4.1) extended FaF [61] by predicting the intent which was defined as the combination of the target high-level behavior (e.g. moving directions) and motion trajectory. Besides SPL input, the authors took an extra rasterized map as network's input. The rasterized map consisted of the binary mask and poly lines which encoded static scene information including roads, traffic lights, traffic signs, etc. These signals provided a strong motion prior and contributed a lot to the intent prediction. The study [136] further extended IntentNet [12] to integrate motion planning into the end-to-end motion forecasting system. Instead of just predicting the moving angle as IntentNet [12], the purpose of motion planning was to generate one optimistic trajectory with minimum cost. Note that due to the novel joint design, multi-modality models were trained together in an end-to-end manner. The proposed motion planning was interpretable and generalized well to the uncertain situation. The [11] was also developed based on IntentNet [12] by adding the interaction model at the end for motion predictions. It exploited a graph-based convolution neural network to model the relation between various actors and further decide the trajectory according to probabilistic inference.

Nevertheless, these methods are all developed following the object detection-tracking-forecasting schema. The performance of the motion forecasting inevitably depend on the accuracy of bounding box positions derived from the first detection stage. If there are some unexpected objects failed to be detected or some unseen objects which are pretty normal in the real traffic situation, final forecasting results will be affected.

7.1.2 OGM Representation

Occupancy grid map (OGM) was another popular representation for point cloud data. It partitioned the space into 2D grid cells with each cell indicating the occupancy and the point velocity of the space. The occupancy representation helped to predict the existence confidence of objects and thus did not need bounding boxes as the detection results. Schreiber et al. [91] was the one that adopted the occupancy grid map to forecast future motion for sequential raw sensor data. It converted point cloud frames to a sequence of

TABLE 12: The summary of the sequential point cloud forecasting methods.

Methods			Code	Attribute
Motion Forecasting	BEV Representation	FaF [61]	✓	The BEV Representation is more convenient to implement due to the regular projection which also makes the network more efficient
		IntentNet [12]	✓	
		Spagnn [11]	✓	
		NMP [136]	×	
	OGM Representation	Schreiber et al. [91]	×	The OGM Representation release the dependence on the object detection results and improve the generalization ability.
		MotionNet [123]	✓	
	Range View Representation	LaserFlow [69]	✓	Preserves more information from the raw point clouds.
Sequential Forecasting	Single-frame prediction	Sun et al. [100]	×	These two methods are limited to the single frame future prediction instead of sequential forecasting
		Deng et al. [17]	✓	
	Multi-frames prediction	Weng et al. [118]	✓	The methods are adopting the range-view representation. Sun et al. [100] and Mersch et al. [67] are limited to the deterministic prediction while S2net [116] explore to extend the future uncertainty prediction.
		Mersch et al. [67]	✓	
		S2net [116]	×	

TABLE 13: Quantitative detection and motion forecasting results on the NuScenes dataset.

Method	Average Precision (%)	L_2 Error (cm)			Classification Accuracy (%)	
	0.7 IoU	0.0 s	1.0 s	3.0 s	MCA (Mean Category Accuracy)	OA (Overall Accuracy)
Schreiber et al. [91]	-	-	-	-	69.6	92.8
MotionNet [123]	-	-	-	-	70.3	95.8
SpAGNN [11]	-	**22**	58	145	-	-
LaserFlow [69]	56.1	25	**52**	**143**	-	-

TABLE 14: Quantitative detection and motion forecasting results on the ATG4D dataset.

Method	Average Precision (%)	L_2 Error (cm)		
	0.7 IoU	0.0 s	1.0 s	3.0 s
FaF [61]	64.1	30	54	180
IntentNet [12]	73.9	26	45	146
NMP [136]	80.5	23	36	114
SpAGNN [11]	83.9	22	33	**96**
LaserFlow [69]	**84.5**	**19**	**31**	99

dynamic occupancy grid maps and input them to a ConvLSTM encoder-decoder network to capture temporal dependencies. The ConvLSTM could predict future dynamic objects separating with the static scene. The authors added skip connections to the RNN network capturing multi-resolution features which could enhance the performance of the small object prediction.

However, one major problem of the occupancy grid representation is hard to find the temporal correspondence between cells, which could further prevent better modeling behavior relations. Besides this, it also excludes object class information and sets the barrier for deeper analysis of the forecasted motion. Thus, MotionNet [123] combined BEV and occupancy map representations and devised a novel representation named BEV map. It extended from the OGM and enriched the representation including the occupancy, motion, and object category information. After converting point cloud frames to a sequence of BEV maps, they were sent into MotionNet to obtain the scene perception and predict motion information. Specifically, MotionNet exploited a novel spatio-temporal pyramid network named STPN to extract hierarchical features and jointly modeled the space-time relations. Meanwhile, light block spatio-temporal convolution (STC) was developed to reduce computation cost of high dimension data and achieve real-time running.

7.1.3 Range View Representation

Though two representations mentioned above could achieve promising performance for motion forecasting, they still suffer from quantization error and lose the information during the compression process. LaserFlow [69] proposed to use the range view representation which provided more information than the BEV representation. As we also introduced in previous sections, the range map comes from spherical projection of point clouds. LaserFlow [69] treated multiple frames of range maps produced from point clouds as the input of the network. To aggregated multiple range maps, the multi-sweep fusion architecture was proposed to solve the coordinate system dis-alignment problem. In addition to extracting range map features, the authors exploited a transformer sub-network to unify the coordinate system and align all of the sweeps' features to the current one. The follow-by object detection and motion prediction network was applied to complete the motion forecasting by utilizing uncertainty curriculum learning.

7.2 Sequential Pointcloud Forecasting

The SPF (Sequential Pointcloud Forecasting) task is defined to predict future M point cloud frames given previous N frames. Instead of forecasting future point cloud information on the object level, SPF predicts the whole scene point clouds including foreground objects and background static scene. Also different from other generation tasks such as [21], [52] mostly inferring the single point cloud frame [100], SPF forecasts a sequence of future point cloud frames which requires longer temporal range information and more holistic scene understanding. Figure 11 demonstrates the difference between the motion forecasting and the sequential forecasting pipelines.

Paper [100] aimed to resolve the point cloud compression and remove the redundancy part of spatial and temporal domains. It devised a ConvLSTM structure to predict future point cloud frames instead of using the 1D LSTM in [118]. Deng et al. [17] proposed a learning schema which adopted the scene flow embedding [56] to model the temporal relation among four input point cloud frames. PointNet++ [81] and Edge Conv [111] were introduced to extract 3D spatial features. Combining spatial and temporal features, the network output the next future frame. However, the methods proposed by Sun et al. [100] and Deng et al. [17] were limited to single future frame prediction setting while SPF requires a sequence of frames as inference results.

Weng et al. [118] firstly investigated the SPF (Sequential Pointcloud Forecasting) task and proposed a delicate method

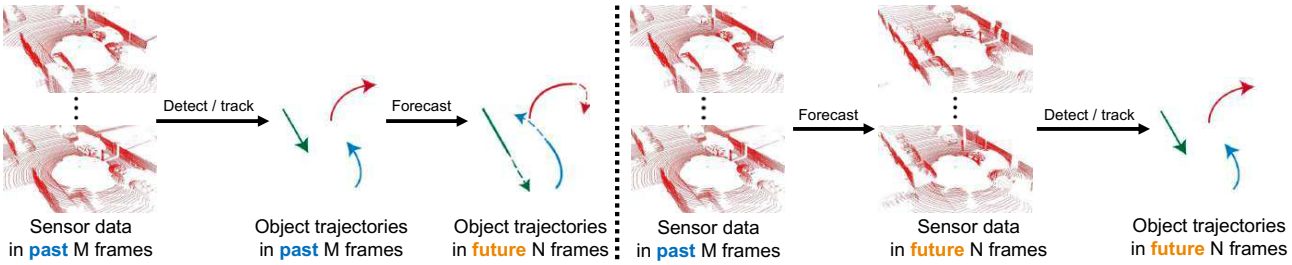


Fig. 11: Comparison of motion forecasting and sequential forecasting pipeline. The figure is from [118] with author's permission.

SPFNet which was able to predict the entire future 3D scene regardless of the human annotated ground-truth trajectories. The way it achieved this goal was through devising a novel forecast-then-detect schema to replace the conventional detect-then-forecast idea. In this way, all of the signals for the network training were future point cloud frames in a self-supervised manner. The proposed SPFNet employed the range map-based encoder and decoder structure to generate future point clouds. Meanwhile, a sequence of LSTMs was adopted to model the temporal relation among point cloud frames. The authors also exploited a new evaluation protocol that connected the detection and forecasting performance together to better assess the model. The SPFNet achieved the state-of-the-art performance on benchmark datasets compared to previous detect-then-forecast pipelines.

Instead of leveraging the LSTM structure, Mersch et al. [67] proposed to utilize the 3D convolution to jointly learn spatial-temporal features of input point cloud sequences. It converted point clouds to range images which were then sent to an encoder-decoder network structure to extract features. Meanwhile, Skip Connections and Horizontal Circular Padding was introduced to capture detailed spatial-temporal information. Finally, the predicted future range images were converted back to sequential point clouds as output.

7.3 Discussion

Tables 13 and 14 summarize results of motion forecasting on ATG4D and NuScenes datasets respectively. The observations and discussed can be found as follows:

- Though BEV representation is more frequently used, the methods adopting range view representation achieve better performance due to more complete information embedded.
- Though existing motion forecasting methods have achieved remarkable performance on benchmarks, the errors sharply increase when the time range is extended. This shows the limitation for handling longer-range SPL data.

8 Future Directions

Sequential point clouds have been attracting great attention due to the need for a better and holistic scene understanding. Many methods have demonstrated the efficacy for processing high dimension data but with challenges and limitations. This section discusses some potential future research directions on the sequential point clouds.

Longer-range temporal dependency Spatial feature learning has made great progress. The way how to capture and address temporal information is crucial for spatio-temporal learning. The existing research of sequential point clouds has attempted to model the temporal relation and leverage the long-range dependency to various applications such as tracking and forecasting. However, it is usually difficult to be accurate when the time range increases no matter for the input sequence or the output sequence. Another issue for a longer time range is the expensive computation cost due to a large amount of data. One possible solution is to exploit point cloud compression techniques such as utilizing flow information to fill the temporal gaps. Meanwhile, transformers have been approved to be quite good at modeling temporal attention and capturing long-range dependencies. Therefore, the combination of the two ideas could be an exciting future direction to model longer-range temporal dependencies.

Multitask Learning Holistic perception of a scene is the foundation for applications of the sequential point clouds. Various tasks such as scene flow estimation, object detection and tracking, as well as segmentation, play an important role. For instance, scene flow estimation could provide the motion status of surrounding objects, while segmentation could deliver the object category information. However, by simply conducting these tasks separately, none of them could provide holistic guidance, while the results between tasks might even be inconsistent. Thus one possible solution is to jointly learn those essential features (e.g. semantic flow) across multitasks. For example, unified architectures could be designed to simultaneously learn scene flow and segmentation. The learned scene flow features and semantic features could associatively boost each other while keeping the temporal consistency along the time sequence. Other multitask learning schemas are also worth devising especially for complex high dimensional data.

Generative Models Recent advances in sequential point cloud learning have been significantly driven by generative models, especially with the integration of point cloud implicit representations. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) designed for point cloud data are now being merged with temporal architectures such as Long Short-Term Memory (LSTM) networks. This integration facilitates the capture of dynamic behaviors within sequences of point clouds.

In the realm of generative models like GANs and VAEs, implicit neural embeddings enable these networks to generate highly detailed and complex 3D shapes with greater precision. The neural network can implicitly model the intricate geometrical relationships within the 3D space, allowing for the creation of shapes that are challenging to achieve with explicit representations. This capability is particularly beneficial in fields like biomedical imaging, architectural design, and 3D animation, where accuracy and detail are paramount. Incorporating these embeddings into temporal architectures like LSTM networks leads to more advanced applications in dynamic 3D data processing. For example, in sequential point cloud data, such as those captured by LiDAR sensors in autonomous vehicles or 3D motion capture systems, implicit neural embeddings can track and predict complex changes in the 3D shapes over time, facilitating advanced motion prediction

and temporal scene understanding.

Furthermore, transformer-based architectures, renowned for their sequence modeling capabilities in natural language processing, are being adapted to process temporal point cloud sequences. By incorporating implicit representation techniques, these architectures can provide enhanced attention mechanisms and context-aware representations. This adaptation is pivotal for tasks like anomaly detection and event segmentation in dynamic 3D environments, as it allows for a more nuanced understanding of the spatial-temporal interplay within point cloud data.

Large Language Models Drawing from the strengths of LLVMs in bridging text and visuals, we could devise algorithms that combine sequential point cloud data with added elements like annotations or descriptions. Leveraging the attention mechanisms inherent in transformer models, this approach provides a richer insight into the evolving dynamics of point cloud sequences. This advancement not only elevates tasks like motion tracking and scene interpretation but also paves the way for generating descriptions of changing 3D visuals. Furthermore, by employing transfer learning strategies commonly associated with LLVMs, these algorithms benefit from vast pretrained datasets, refining their capability to understand sequential point cloud patterns.

9 Conclusion

Deep learning for sequential Deep learning applied to sequential point clouds has achieved significant success in enhancing our understanding of the dynamic world from a spatio-temporal perspective. It has demonstrated remarkable performance across various applications. In this survey, we have offered a comprehensive overview of recent deep learning techniques tailored to the processing of sequential point clouds, along with insights into their application in downstream tasks. We anticipate that this survey will serve as valuable guidance for researchers within the computer vision and multimedia communities, aiding them in their endeavors.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 10

[2] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, pages 753–762. Wiley Online Library, 2010. 5

[3] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Bjorn Ottersten. Deep learning advances on different 3D data representations: A survey. *arXiv preprint arXiv:1808.01462*, 2018. 2

[4] Mehmet Aygün, Aljoša Ošep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4d panoptic lidar segmentation. *arXiv preprint arXiv:2102.12472*, 2021. 13

[5] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for 3d scene flow estimation from point clouds. *arXiv preprint arXiv:1806.02170*, 2018. 4, 5

[6] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 13, 14

[7] Jens Behley, Andres Milioto, and Cyrill Stachniss. A benchmark for lidar-based panoptic segmentation based on kitti. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13596–13603. IEEE, 2021. 13

[8] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection, 2020. 8

[9] 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. 8

[10] Hongxiang Cai, Zeyuan Zhang, Zhenyu Zhou, Ziyin Li, Wenbo Ding, and Jiuhua Zhao. Bevfusion4d: Learning lidar-camera fusion under bird's-eye-view via cross-modality guidance and temporal aggregation. *arXiv preprint arXiv:2303.17099*, 2023. 7, 8, 9

[11] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9491–9497. IEEE, 2020. 14, 15

[12] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018. 7, 14, 15

[13] Guoqing Chen, Xingchao Yu, Yujun Chen, Ziyi Yao, Junfei Xie, and Jiashi Feng. Sparse attentive equivariant graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[14] Lin-Zhuo Chen, Xuan-Yi Li, Deng-Ping Fan, Ming-Ming Cheng, Kai Wang, and Shao-Ping Lu. LSANet: Feature learning on point sets by local spatial attention. *arXiv preprint arXiv:1905.05442*, 2019. 12

[15] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3d multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*, 2020. 10

[16] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 12, 13

[17] David Deng and Avideh Zakhor. Temporal lidar frame prediction for autonomous driving. In *2020 International Conference on 3D Vision (3DV)*, pages 829–837. IEEE, 2020. 15

[18] Fabian Duerr, Mario Pfaller, Hendrik Weigel, and Jürgen Beyerer. Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In *2020 International Conference on 3D Vision (3DV)*, pages 781–790. IEEE, 2020. 12, 13

[19] Ahmad El Sallab, Ibrahim Sobh, Mahmoud Zidan, Mohamed Zahran, and Sherif Abdelkarim. Yolo4d: A spatio-temporal approach for real-time multi-object detection and classification from lidar point clouds. *NIPSW*, 2018. 7, 8

[20] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know what your neighbors do: 3D semantic segmentation of point clouds. In *ECCVW*, 2018. 12

[21] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 6, 15

[22] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2021. 12, 13

[23] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, volume 33, pages 3558–3565, 2019. 3

[24] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642. IEEE, 2018. 10

[25] Jingyun Fu, Zhiyu Xiang, Chengyu Qiao, and Tingming Bai. Pt-flownet: Scene flow estimation on point clouds with point transformer. *IEEE Robotics and Automation Letters*, 8(5):2566–2573, 2023. 5, 6

[26] Kent Fujiwara and Taiichi Hashimoto. Neural implicit embedding for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11734–11743, 2020. 3

[27] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection, 2020. 8

[28] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3D siamese tracking. *arXiv preprint arXiv:1903.01784*, 2019. 10

[29] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014. 12

[30] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 12

[31] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. HPLFlowNet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *CVPR*, pages 3254–3263, 2019. 5

[32] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *ArXiv*, abs/1912.12033, 2019. 1, 2

[33] Chunyong Hu, Hang Zheng, Kun Li, Jianyun Xu, Weibo Mao, Maochun Luo, Lingxuan Wang, Mingxia Chen, Kaixuan Liu, Yiru Zhao, et al. Fusionformer: A multi-sensory fusion in bird's-eye-view and temporal consistent transformer for 3d objection. *arXiv preprint arXiv:2309.05257*, 2023. 7, 8, 9

[34] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020. 7, 8, 9

[35] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. *arXiv preprint arXiv:1911.11236*, 2019. 12

[36] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675. IEEE, 2016. 11

[37] Rui Huang, Wanyue Zhang, Abhijit Kundu, C. Pantofaru, David A. Ross, T. Funkhouser, and A. Fathi. An lstm approach to temporal 3d object detection in lidar point clouds. In *ECCV*, 2020. 7, 8

[38] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 13

[39] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 5

[40] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *CSUR*, 2017. 2

[41] Varun Jampani, Martin Kiefel, and Peter V Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4452–4461, 2016. 5

[42] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2020. 3

[43] Mingyang Jiang, Yiran Wu, and Cewu Lu. PointSIFT: A sift-like network module for 3D point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018. 12

[44] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world. *arXiv preprint arXiv:2103.01306*, 2021. 4, 5

[45] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5010–5019, 2018. 3

[46] Mohammad Khoury, Marek Rajchl, Matt McCann, Andreas Geiger, and Laura Leal-Taixe. D2feat: Learning to match keypoints with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[47] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020. 13

[48] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 13

[49] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. *arXiv preprint arXiv:1812.05784*, 2018. 4, 7, 8

[50] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1254–1262, 2022. 5, 6

[51] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *NeurIPS*, pages 820–830, 2018. 3

[52] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction. In *AAAI*, 2018. 15

[53] Jiayuan Lin, Yifan Wu, Jiwen Lu, and Jiaya Jia. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[54] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[55] Weiping Liu, Jia Sun, Wanyi Li, Ting Hu, and Peng Wang. Deep learning on point clouds and its application: A survey. *Sensors (Basel, Switzerland)*, 19, 2019. 2

[56] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3D: Learning scene flow in 3D point clouds. In *CVPR*, pages 529–537, 2019. 4, 5, 15

[57] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep learning on dynamic 3D point cloud sequences. In *ICCV*, 2019. 5, 12, 13

[58] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, pages 1–10, 2019. 3

[59] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for efficient 3D deep learning. In *NeurIPS*, pages 963–973, 2019. 1

[60] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 11

[61] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, pages 3569–3577, 2018. 7, 10, 11, 14, 15

[62] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, September 2015. 3

[63] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 5

[64] Scott McCrae and A. Zakhor. 3d object detection using temporal lidar data. In *Arxiv*, 2020. 7, 8

[65] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. *ArXiv*, abs/1811.04337, 2018. 11

[66] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 5

[67] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. *arXiv preprint arXiv:2110.04076*, 2021. 15, 16

[68] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 3

[69] Gregory P Meyer, Jake Charland, Shreyash Pandey, Ankit Laddha, Shivam Gautam, Carlos Vallespi-Gonzalez, and Carl K Wellington. Laserflow: Efficient and probabilistic object detection and motion forecasting. *IEEE Robotics and Automation Letters*, 6(2):526–533, 2020. 14, 15

[70] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. 11, 13

[71] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11177–11185, 2020. 4, 5, 13

[72] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, Zhifeng Chen, Jonathon Shlens, and Vijay Vasudevan. Starnet: Targeted computation for object detection in point clouds. *CoRR*, 2019. 8

[73] nuTonomy. nuscenes 3d object detection challenge. https://www.nuscenes.org/object-detection?externalData=all&mapData=all&modalities=Any/. 8

[74] Pablo R Palafox and Matthias Niessner. Voxflownet: Learning scene flow in 3d point clouds through voxel grids. *ArXiv*, 2017. 4, 5

[75] Ziqi Pang, Zhichao Li, and Naiyan Wang. Model-free vehicle tracking and state estimation in point cloud sequences. *arXiv preprint arXiv:2103.06028*, 2021. 11

[76] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 3

[77] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 15435–15445, 2021. 7

[78] C. Qi, Y. Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. *ArXiv*, abs/2103.05073, 2021. 8, 9

[79] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. *arXiv preprint arXiv:1904.09664*, 2019. 10

[80] Charles R. Qi, H. Su, Kaichun Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 1, 3, 5, 12

[81] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 1, 3, 4, 5, 12, 13, 15

[82] Haozhe Qi, C. Feng, Zhiguo Cao, F. Zhao, and Yang Xiao. P2b: Point-to-box network for 3d object tracking in point clouds. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6328–6337, 2020. 10, 11

[83] Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, and Ke Lu. Recent advances in 3D object detection in the era of deep neural networks: A survey. *IEEE TIP*, 2019. 2

[84] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 8

[85] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *ECCV*, pages 596–611, 2018. 11

[86] Gernot Riegler, Ali O. Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6620–6629, 2016. 12

[87] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 12

[88] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv preprint arXiv:1912.05905*, 2019. 11

[89] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 3

[90] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. In *PBG@ SIGGRAPH*, pages 111–120, 2006. 12

[91] Marcel Schreiber, Stefan Hoermann, and Klaus Dietmayer. Long-term occupancy grid prediction using recurrent neural networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9299–9305. IEEE, 2019. 14, 15

[92] Lin Shao, Parth Shah, Vikranth Dwaracherla, and Jeannette Bohg. Motion-based object segmentation based on dense rgb-d scene flow. *IEEE Robotics and Automation Letters*, 3(4):3797–3804, 2018. 4

[93] Hanyu Shi, Guosheng Lin, Hao Wang, Tzu-Yi Hung, and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 12, 13

[94] 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. 8

[95] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Sämann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds. *arXiv preprint arXiv:1904.07537*, 2019. 10

[96] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 8

[97] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *CVPR*, 2018. 11

[98] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 5

[99] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 4, 8

[100] Xuebin Sun, Sukai Wang, Miaohui Wang, Zheng Wang, and Ming Liu. A novel coding architecture for lidar point cloud sequence. *IEEE Robotics and Automation Letters*, 5(4):5637–5644, 2020. 15

[101] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, 2018. 13

[102] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In *3DV*, pages 537–547, 2017. 11

[103] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. *arXiv preprint arXiv:1904.08889*, 2019. 3, 12, 13

[104] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 12

[105] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019. 13

[106] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3D object recognition. *arXiv preprint arXiv:1906.01592*, 2019. 3

[107] Guangming Wang, Yunzhe Hu, Xinrui Wu, and Hesheng Wang. Residual 3-d scene flow learning with context-aware feature extraction. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2022. 5, 6

[108] Haiyan Wang, Jiahao Pang, Muhammad A. Lodhi, Yingli Tian, and Dong Tian. Festa: Flow estimation via spatial-temporal attention for scene point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14173–14182, June 2021. 4, 5

[109] Sukai Wang, Yuxiang Sun, Chengju Liu, and Ming Liu. Pointtracknet: An end-to-end network for 3-d object detection and tracking from point clouds. *IEEE Robotics and Automation Letters*, 5(2):3206–3212, 2020. 10, 11

[110] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Thomas Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020. 8

[111] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 15

[112] Zirui Wang, Shuda Li, Henry Howard-Jenkins, Victor Adrian Prisacariu, and Min Chen. FlowNet3D++: Geometric losses for deep scene flow estimation. *arXiv preprint arXiv:1912.01438*, 2019. 4, 5

[113] Zongji Wang and Feng Lu. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. *IEEE transactions on visualization and computer graphics*, 2019. 1

[114] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. PV-RAFT: Point-Voxel Correlation Fields for Scene Flow Estimation of Point Clouds. In *CVPR*, 2021. 4, 5

[115] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019. 13

[116] Xinshuo Weng, Junyu Nan, Kuan-Hui Lee, Rowan McAllister, Adrien Gaidon, Nicholas Rhinehart, and Kris Kitani. S2net: Stochastic sequential pointcloud forecasting. *https://www.xinshuoweng.com/papers/S2Net/arXiv.pdf*, 2021. 15

[117] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. *arXiv preprint arXiv:1907.03961*, 2020. 10, 11

[118] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. *arXiv preprint arXiv:2003.08376*, 2020. 15, 16

[119] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Unsupervised sequence forecasting of 100,000 points for unsupervised trajectory forecasting. *arXiv e-prints*, pages arXiv–2003, 2020. 1

[120] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020. 10, 11

[121] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional neural nets with recurrent crf for real-time road-object

segmentation from 3D lidar point cloud. In *ICRA*, pages 1887–1893, 2018. 11

[122] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, pages 4376–4382, 2019. 11

[123] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11385–11395, 2020. 14, 15

[124] Wenxuan Wu, Li Fuxin, and Qi Shan. Pointconvformer: Revenge of the point-based convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21802–21813, 2023. 5, 6

[125] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 12

[126] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Fuxin Li. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *ArXiv*, abs/1911.12408, 2019. 5

[127] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 3

[128] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. A review of point cloud semantic segmentation. *arXiv preprint arXiv:1908.08854*, 2019. 2

[129] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19. Springer, 2020. 11

[130] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 7

[131] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. *arXiv preprint arXiv:1904.03375*, 2019. 12

[132] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 7

[133] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 3

[134] Yangyang Ye, Houjin Chen, Chi Zhang, Xiaoli Hao, and Zhaoxiang Zhang. Sarpnet: Shape attention regional proposal network for lidar-based 3d object detection. *Neurocomputing*, 379:53–63, 2020. 8

[135] 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. 7, 8, 9

[136] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019. 14, 15

[137] Yihan Zeng, Da Zhang, Chunwei Wang, Zhenwei Miao, Ting Liu, Xin Zhan, Dayang Hao, and Chao Ma. Lift: Learning 4d lidar image fusion transformer for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17172–17181, 2022. 7, 8, 9

[138] Chris Zhang, Wenjie Luo, and Raquel Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *2018 International Conference on 3D Vision (3DV)*, pages 399–408. IEEE, 2018. 11

[139] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2365–2374, 2019. 10

[140] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 11

[141] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics. *arXiv preprint arXiv:1908.06295*, 2019. 12

[142] Chenxi Zhao, Weihao Zhou, Li Lu, and Qijun Zhao. Pooling scores of neighboring points for improved 3D point cloud segmentation. In *ICIP*, pages 1475–1479, 2019. 12

[143] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, pages 5565–5573, 2019. 12

[144] 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 *Conference on Robot Learning*, pages 923–932, 2020. 8

[145] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1, 4

Haiyan Wang received the B.E. degree from Beijing University of Posts and Telecommunications, China, in 2017, and the Ph.D. degree from The City College, The City University of New York, US, in 2023. His current research focuses on computer vision and deep learning including scene flow estimation, room layout estimation, 3D scene reconstruction, sequential point cloud learning, large language vision models, and 2D & 3D foundation models.

Yingli Tian (M'99–SM'01–F'18) received the B.S. and M.S. degrees from Tianjin University, China, in 1987 and 1990, and the Ph.D. degree from Chinese University of Hong Kong, Hong Kong, in 1996. After holding a faculty position at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University in 1998, where she was a postdoctoral fellow at the Robotics Institute. She then worked as a research staff member in IBM T. J. Watson Research Center from 2001 to 2008. She is one of the inventors of the IBM Smart Surveillance Solutions. Currently she is a CUNY Distinguished Professor in the Department of Electrical Engineering at the City College and the Department of Computer Science at the Graduate Center, the City University of New York. Her research focuses on a wide range of computer vision problems from scene understanding, medical imaging analysis, human behavior analysis, to facial expression recognition and assistive technology. She is a fellow of IEEE and IAPR.